CPU

5,50 DM
6,00 sFr
48 öS

März 1985 Nr. 3

Das kompetente Anwender-Magazin

CPU-Game

- Wieder Spitzenspiele für Ihren Home-computer

CPU-Markt

- Informationen
- Trends
- Tests

CPU-Utility

Ia Anwender-Software:

Monitor für TI-99/C 64

MC-Routinen für ZX-Spectrum

Zeichensatz – Konverter für MSX

und vieles mehr

CPU-Happy

wieder ein kniffliges Rätsel zum Mitmachen

NEU!!! Jetzt auch Programme für MSX und CPC-464

Liebe Leser der „CPU“!

Wir haben im Monat Januar eine Vielzahl von Anfragen erhalten, die sich auf das Ausbleiben der „CPU“ (1./85) beziehen. Hier die Antwort:
Der Tronic-Verlag hat um die Jahreswende 84/85 die Titelrechte der „CPU“ übernommen. Da eine Frist von ca. 2 Wochen bei weitem nicht ausreicht, um eine Zeitschrift wie die „CPU“, den Bedürfnissen der Leser entsprechend, zusammenzustellen, entschloß sich die Redaktion, die neuste Ausgabe erst im Februar zu veröffentlichen.

Um so mehr hoffen wir, daß die Ihnen vorliegende März-Ausgabe unserem Ziel, auch 1985 wieder das Beste für Sie zusammenzutragen, ein Stück näher gerückt ist. Haben Sie diesbezüglich die gleiche Meinung?
Haben Sie Grund zur Kritik oder Fragen?
Schreiben Sie oder rufen Sie uns an!!

Noch eine Ergänzung zum Verlagswechsel:

Zwischenzeitlich erhielten wir trotz des Hinweises in unserer letzten Ausgabe wieder viele Anrufe über den Verbleib eingesandter Manuskripte und Datenträger oder bestellter Software, die dem bisherigen Verlag der CPU zugesandt bzw. dort bestellt wurde.

Wir bitten Sie, in diesem Zusammenhang zu bedenken, daß der Tronic-Verlag lediglich die Titelrechte, nicht aber den bisherigen Verlag der Zeitschrift CPU erworben hat. Unser Haus trägt keinerlei Verantwortung für Reklamationen, nicht gelieferte Ausgaben bei Abonnements etc., die aus dem Jahr 1984 datieren!

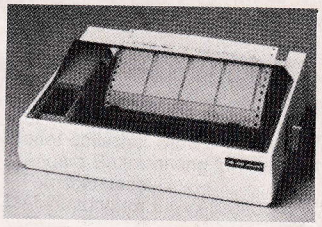
Wir machen weiterhin darauf aufmerksam, daß jegliche Post ausschließlich an unseren Verlag, nicht an das Druckhaus oder den Vertrieb (Verlagsunion Wiesbaden) gerichtet werden muß.

Eine Ausnahme hierzu bildet lediglich der Abo-Schriftverkehr, der direkt an die Verlagsunion Wiesbaden, Postfach 5707, Friedrich-Bergius-Straße 7, gerichtet wird.

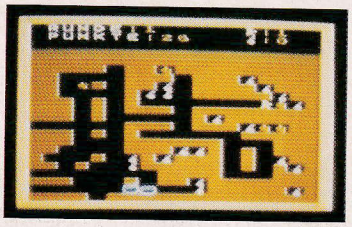
Die Redaktion

CPU März '85

Spritehilfe für C-64
Mit Hilfe dieses Programms werden Bewegungen von Sprites ermöglicht, ohne den Ablauf des Hauptprogramms zu beeinflussen. S. 29



Drucker unter 1300,– DM
Welche gibt es und was leisten sie. S. 75



Cherry-Harry
Wieder ein tolles Spiel für den Atari, das einiges an Geschick erfordert. S. 14

CPU-Utility

MSX

Zeichensatz-Konverter 9

ZX-Spectrum

MC-Routinen 20

Data-Base 24

Pixel 28

TI-99

Monitor 56

Funktionsanalyse 64

Hardcopy 61

Atari

RAM-Tester 11

C-64

Spritehilfe 29

Diskloader 42

Direktory 43

Datagenerator 46

Tape-Directory 48

ASMON 37

Apple

Dateiverwaltung 50

CPU-Markt

Neues vom Computermarkt 4/74

CPU-Game

Atari

Gnom-Painter 12

Cherry-Harry 14

CPC-464

Autorennen 17

Rubriken

CPU-Wörterbuch 8

Impressum 74

CPU-Harry 70

CPU-Bibliothek 76

Softwareservice 78

Mitmachen – Softwareautoren gesucht 77

Datenverarbeitung im Zeitraffer 63

Computer-Messen 1985 72

CAD im Kunststoffbetrieb

Ihre aktuelle Informationsveranstaltung über CAD wiederholen das Institut für Kunststoffverarbeitung (IKV) in Aachen und der Gesamtverband Kunststoffverarbeitende Industrie (GKV) am 7. März. Angesprochen sind die Entscheidungsträger von Unternehmen, die im Bereich der Auslegung und Konstruktion von Formteilen und Werkzeugen arbeiten, d. h. in den Verfahren Spritzgießen von Thermoplasten, Duroplasten (auch kurzfaserverstärkte) sowie Elastomeren, Preß-, SMC- und RIM-Verfahren. Behandelt werden die Themen Anforderungen des Formteilkonstrukteurs an ein CAD-System, Werkzeugkonstruktion und Maschinenprogrammierung, Kopplung von CAD/CAM-Systemen, notwendige Rechnerausstattung sowie Wirtschaftlichkeitsrechnung für eine CAD-Anlage. Auskünfte erteilt das Institut für Kunststoffverarbeitung, Aachen.

1,5 MByte auf einer flexiblen 5-1/4"-Diskette

Der Memorex-Konzern stellt eine flexible 5-1/4"-Diskette mit 60 % mehr Speicherkapazität als auf den derzeit bekanntesten Disketten mit hohen Aufzeichnungsdichten vor.

Diese neue Diskette entspricht den Memorex-Standards und wurde speziell für den europäischen und japanischen Markt entwickelt, auf dem die 8"-Diskettenlaufwerke nun durch steckerkompatible 5-1/4"-Laufwerke ersetzt werden. Aus diesem Grund ist diese 5-1/4"-Memorexdiskette mit einer (unformatierten) Kapazität von 1,6 MByte mit 8"-Disketten der gleichen Kapazität vollkommen kompatibel. Es handelt sich um eine doppelseitige Diskette mit 77 Spuren, die ihre hohe Kapazität durch eine größere Bitdichte pro Spur und eine höhere Koezitivfeldstärke als die normalen 5-1/4"-Disketten erreicht.

Sämtliche flexiblen Disketten von Memorex befinden sich in Hüllen, die in einem Spezialverfahren mit einer durchgehenden Naht geschlossen werden. Dadurch erreichen sie, wie der Hersteller betont, eine größere Datensicherheit und einen besseren Schutz gegen Fremdkörper als sie das Punktnahtverfahren bieten kann.

Memorex, eine Tochtergesellschaft des Burroughs-Konzern, produziert und vertreibt weltweit Datenspeicheranlagen, Kommunikationsanlagen und Computerprodukte. Die Media Products Group des Unternehmens produziert und vertreibt Disketten, Magnetbänder und sonstiges Computerzubehör.

Die neuen 5-1/4"-Memorexdisketten mit einer Speicherkapazität von 1,6 MByte können direkt ab Lager bei den lokalen Vertretungen des Unternehmens in verschiedenen Ländern der Welt bestellt werden.

Diskettenseminar auf Video

Als erster Speichermedien-Anbieter stellt die BASF dem EDV- und Büro-Fachhandel Video-Informationen zur Verfügung.

Die BASF-Video-Information-I mit Begleitheft ist ein Diskettenseminar zum Selbststudium. Sie ist in Deutsch und Englisch verfügbar und in Französisch, Spanisch und Niederländisch in Vorbereitung.
Der Videofilm erklärt Fachbegriffe, Typen, Bestellcodes und Kompatibilitätsfragen der 5.25"-Diskette. Das Begleitheft beschreibt darüber hinaus auch alle anderen Diskettentypen und hat ein

Glossarium wichtiger Fachausdrücke des Mikrocomputerbereichs. Ein Suchregister erleichtert das Auffinden aller Begriffe.

Der Musiker ist ein Roboter

Ein menschenähnlicher Roboter, der bei der ab März 1985 in Tsukuba/Japan stattfindenden „Weltausstellung für Technologie und Wissenschaft" die Besucher in der Halle der japanischen Regierung musikalisch unterhalten soll, wartet bereits auf seinen Auftritt. Das 1,80 m große und 90 kg schwere Gerät kann hören, sprechen und sehen und mit seinen beiden Händen und Füßen auf einer elektronischen Orgel spielen. Auf akustischen Befehl werden gelernte Werke vorgeführt. Bei noch nicht gelernten Werken werden die Noten gelesen.

Die Entwicklung dieses Roboters wurde seit etwa eineinhalb Jahren in einer Zusammenarbeit von Sumitomo Electric mit den Wissenschaftlern Kato, Narito, Otel und Shirai der Waseda-Universität in Tokio betrieben. Bei der Konstruktion des Roboters haben sich die Beteiligten um möglichst große Menschenähnlichkeit bemüht. Dazu mußten in Arme, Hände und Beine insgesamt 50 Gelenke eingebaut werden. Das Gerät faßt mehrere Entwicklungen der Waseda-Universität zusammen, zum Beispiel den Waseda Leg 10 R und den Waseda Automatic Manipulator Wam 7.

Wer ist der Schnellste im ganzen Land?

CPU testet die Verarbeitungsgeschwindigkeit von 17 populären Home- und Personalcomputern:
In den Hardwaretests vieler Computerzeitschriften werden normalerweise die speziellen Eigenschaften des zu testenden Computers aufgeführt. Leider wird jedoch nur selten die Verarbeitungsgeschwindigkeit des Rechners mit denen anderer Computer dieser Klasse verglichen. Doch gerade diese Informationen können für den Computerinteressierten ein wichtiges Auswahlkriterium beim Computerkauf werden. CPU testete die Verarbeitungsgeschwindigkeit von siebzehn Home- und Personalcomputern mit Hilfe der folgenden zwei Benchmark-BASIC-Programme (Testprogramme):

BENCHMARK I:

```
10 FOR I=TO 100
20  A=SQR(I)
30  A=1/I
40  A=SIN(I)
50 NEXT I
60 PRINT "FERTIG"
70 END
```

BENCHMARK II:

```
10 FOR I=3 TO 199 STEP 2
20  FOR  J=3  TO  SQR(I)
    STEP 2
30    A=I/J
40    IF INT(A) = A THEN
      70
50  NEXT J
60  PRINT I
70 NEXT I
80 PRINT "FERTIG"
90 END
```

Benchmark I berechnet die Quadratwurzel, den Kehrwert und den Sinus der Zahlen von 1 bis 100.
Benchmark II ermittelt die Primzahlen von 5 bis 199 und gibt diese auf dem Bildschirm aus.
Folgende Zeiten wurden für die jeweiligen Computer gestoppt (Computernamen in alphabetischer Reihenfolge):

Überraschend ist die hohe Geschwindigkeit des BBC-Micros, der alle anderen Rechner überflügelt. Bei uns ist er noch nicht sehr verbreitet, in England gehört er zu den beliebtesten Homecomputern. Auch der neue Schneider CPC 464 kann offensichtlich eine hohe BASIC-Verarbeitungsgeschwindigkeit entwickeln.

Name	Benchmark I	Benchmark II
Acorn BBC	4.1 sek	3.1 sek
Acorn Electron	6.1 sek	4.8 sek
Alphatronic PC	21.5 sek	19.5 sek
Apple II	8.1 sek	8.7 sek
Atari 600	18.4 sek	17.1 sek
Atari 800 XL	18.4 sek	17.1 sek
Casio FP-200	17.7 sek	15.3 sek
Casio FP-1000	8.3 sek	21.2 sek
Commodore C16	7.9 sek	9.3 sek
Commodore C64	9.1 sek	9.7 sek
Commodore VC20	7.7 sek	8.0 sek
IBM PC	4.3 sek	5.3 sek
Laser 2001	5.3 sek	6.5 sek
Schneider 464	4.7 sek	6.5 sek
Sinclair Spect.	17.6 sek	19.0 sek
SVI 318/328	27.9 sek	23.7 sek
TI 99/4A Ex-BASIC	26.8 sek 15,1 sek	31.1 sek 16,9 sek

(Alle Zeitangaben ohne Gewähr)

Zwar ist der „16-Bit"-Rechner TI-99/4A als recht langsam bekannt, überraschenderweise sind jedoch beide Atari-Modelle (600XL und 800XL) ebenfalls keine „Sprinter".
Es wäre sicherlich interessant gewesen, den neuen Sinclair QL (Quantum Leap) mit in den Vergleichstest zu nehmen; leider stand jedoch noch kein Testgerät zur Verfügung. Auch die MSX-Rechner aus Fernost konnten nicht berücksichtigt werden. (O.S, K.S)

Textverarbeitung für den Hausgebrauch

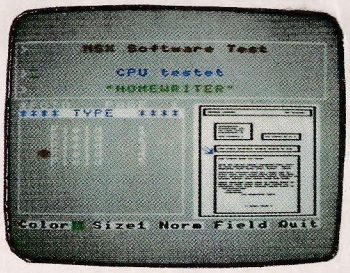
CPU testet den „HOMEWRITER" für MSX-Computer

HOMEWRITER ist ein Textverarbeitungsprogramm, welches auf allen MSX-Computern lauffähig ist.
Das ca. 149,– DM teure Programm wird von SONY auf Modul angeboten. Dadurch entfällt die Ladezeit für das Programm. Das Cartridge wird in den Cartridgeslot gesteckt, und anschließend der Computer eingeschaltet. Danach meldet HOMEWRITER sofort seine Einsatzbereitschaft.
HOMEWRITER unterstützt einen Cassettenrecorder als Speicher, und einen MSX Plotter oder Drucker als Ausgabeeinheit.
Bei Verwendung eines Drukkers muß man jedoch auf einige Funktionen verzichten.

Der Bildschirm wird bei HOMEWRITER in drei Felder aufgebaut. Das erste Feld liegt im oberen Bildschirmbereich über die gesamte Bildschirmbreite. Dieses Feld besteht aus 6 Zeilen, worin der gerade geschriebene Textauszug dargestellt wird. Das zweite Feld liegt im linken unteren Bildschirmbereich, und wird als Menüfeld benutzt, worin mittels der Cursor-Tasten eine entsprechende Auswahl getroffen werden kann. Rechts neben diesem Feld ist Feld 3 aufgebaut. Dieses Feld zeigt in verkleinerter Form das Schriftstück, woran Sie gerade schreiben.

In diesem Schriftstück werden die Wörter als schwarze und die Leerstellen als weiße Striche dargestellt. Auf diese Weise sieht man sofort, wo man sich gerade auf dem Papier befindet, und kann somit die Formatierung des Schriftstückes sehr exakt bestimmen.

HOMEWRITER bietet 5 verschiedene Schriftstückgrößen an: A4, B5, A5, Endlosformat für 114 mm breite Rollen und Postkartengröße. Die Papiergröße wird im Menüfeld mittels der Cursortasten angewählt.



Innerhalb eines Schriftstükkes können bis zu 8 verschiedene Felder festgelegt werden. So kann z. B. ein Feld für die eigene Anschrift, das zweite Feld für die Adresse, das dritte Feld für den Schrifttext usw. bestimmt werden. Die Feldformartierung geht sehr einfach vor sich: Man wählt über die Funktionstasten die Funktion FORMAT. Jetzt wird das entsprechende Feld angewählt, die linke Eckmarke mittels der Cursortasten auf dem Bildschirm an die entsprechende Stelle bewegt, und mittels CORNER diese Position fixiert. Danach wird die rechte Eckmarke auf die gleiche Weise festgelegt und das Feld mittels SET angelegt.
Mittels der Anweisung FIELD kann die Größe bereits erstellter Felder geändert werden, wobei der Text in den Feldern nicht verloren geht.
Leider können einmal gesetzte Felder nicht wieder gelöscht werden.

Diese Formatierung funktioniert jedoch beim Ausdruck nur mit einem Printer Plotter.

Ebenfalls kann beim Einsatz eines Printer Plotters eine von 4 Farben angewählt werden. Diese Farben können auch innerhalb eines Schriftstückes gewechselt werden.

Als weitere Sonderfunktion eines Printer Plotters kann zwischen 7 Schriftgrößen und 3 verschiedenen Zeilenabständen gewählt werden.

Für den Ausdruck stehen 2 verschiedene Funktionen zur Verfügung: PLOT gibt den Text über einen Printer Plotter aus und PRINT gibt den Text über einen Drucker aus.

HOMEWRITER ist für den privaten Gebrauch gedacht. Für den kommerziellen Einsatz ist HOMEWRITER jedoch kaum geeignet. HOMEWRITER wird dem MSX Motto „Kinderleicht" voll gerecht. Die Anleitung wird derzeit nur in Englisch mitgeliefert, ist jedoch fast „überflüssig", da das Programm durch seine Menüsteuerung sowie durch die Funktionstastenauswahl sehr benutzerfreundlich gestaltet ist.
Durch ein wenig probieren versteht jeder bald auch ohne Anleitung, wie HOMEWRITER zu bedienen ist.

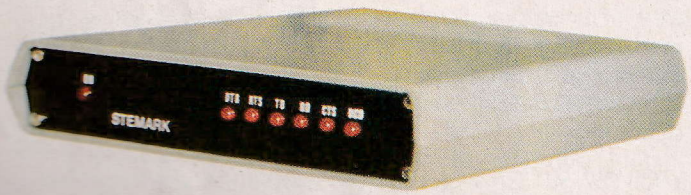
Der Privatmann kann mittels HOMEWRITER zu einem relativ günstigen Preis ein Programm erwerben, mit dessen Hilfe er problemlos seine kompletten Schreibarbeiten erledigen kann.

Für den Privatgebrauch ist HOMEWRITER deshalb „empfehlenswert".

Volker Becker

Modem, das Computer-Kommunikationsmittel

Modems setzen sich bei den Computerfans immer mehr durch, die mit der allgemein üblichen Arbeit am Computer nicht mehr zufrieden sind, und darum nach Kommunikation mit anderen Computern und deren Benutzern streben. Stemark Elektronik entwickelte ein Modem (AS/3/6/12), welches die drei wählbaren Übertragungsgeschwindigkeiten 300 bps full duplex, 600 bzw. 1200 bps half duplex ermöglicht. Das Modem entspricht der CITT-V.21-Norm und ist auto-answer-fähig. **Stemark Elektronik GmbH, Grazer Gasse 35, A-8430 Leibnitz**



Commodore Floppy zu langsam???

Wer sich der Trägheit seiner Commodore 1541 Floppy entledigen möchte, sollte sich die von der Firma Data Becker (Düsseldorf) herausgebrachte Hardwareerweiterung „Floppy Speed" anschaffen. Mit diesem Zusatz und der dazugehörigen Software wird Ihre 1541 bis zu zehnmal schneller. **Data Becker GmbH, Merowingerstr. 30, 4000 Düsseldorf, Tel.: 02 11 / 3 10 01 24**

Ein praktisches Zubehör für den Schneider CPC 464:

Der Y-Adapter für Joysticks von Dynamics.
Der neue **Schneider CPC 464** wird in Fachkreisen und vom Anwender unumstritten positiv beurteilt. Ein kleines Manko ist allerdings, daß das Gerät nur einen Ausgang für Joysticks besitzt. Dadurch ist man gezwungen, um auch zu zweit spielen zu können, die Schneider-Joysticks zu nehmen, wobei aber der 2. Joystick an den ersten angeschlossen wird, was die Bewegungsfreiheit beim Spiel beeinträchtigt.
Mit dem Y-Adapter für Joysticks kann man beide Joysticks direkt am Gerät anschließen, so daß beide Spieler ihre volle Bewegungsfreiheit voneinander erlangen.

Außerdem können mit dem Y-Joystickadapter von Dynamics alle Steuerknüppel mit dem normalen Stecker angeschlossen werden, so daß man auf jeden Fall mit seinen Lieblingsjoysticks auch am Schneider CPC 464 spielen kann.

Den Y-Joystickadapter für den Schneider CPC 464 gibt es ab sofort im Fachhandel zum günstigen Endverbraucherpreis von 24,50 DM.

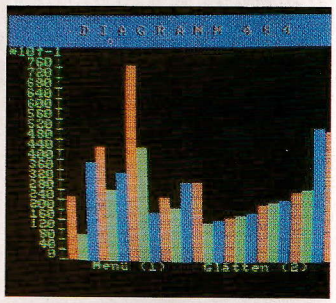
„Diagramm 464"

Oft werden Zahlenwerte wie z. B. von statistischen Erhebungen, Bilanzen oder Monatsumsätzen erst durch die bildliche Umsetzung in ihrer Bedeutung sichtbar und übergreifend interpretierbar.

Mit dem Programm „Diagramm 464" für den **Schneider CPC 464** können selbst eingegebenen Zahlenwerte oder, aus anderen Programmen übernommene Daten auf dem Bildschirm graphisch dargestellt werden.

Zwischen 3 verschiedenen Darstellungsweisen kann man wählen: Balkendiagramm, Strichdiagramm und Funktionskurve.

Die Anzahl der Werte, der Anfangswert und die Position der X-Achse ist einstellbar. Der Anwender hat die Möglichkeit eine Nivellierung vorzunehmen, „Ausreißer" werden erkannt und beseitigt.



Die Werte können abgespeichert und jederzeit wieder geladen werden.

Dynamics bietet das Programm „Graphic Utilities 464" im Rahmen einer neuen, preisgünstigen Softwarereihe für den Schneider CPC 464 an. Diese Reihe besteht aus verschiedenen Archivprogrammen, einem Programm „Graphik Utilities 464" und einem Vokabeltrainer.

Der empf. Endverbraucherpreis für das „Diagramm 464" auf Kassette beträgt 69,– DM.

Hannover-Messe:
Getrennt

Ab 1986 wird CeBIT (Centrum des Büro-, Informations- und Kommunikationstechnik) als eigenständige Verbundmesse veranstaltet. Die CeBIT, bisher Bestandteil der Hannover-Messe, findet in der Zeit vom 12. bis 19. März 1986 statt.
Deutsche Messe und Ausstellungs-AG, Messegelände, 3000 Hannover 82, Tel. 05 11 / 8 91.

Homeword-Textverarbeitung für C 64

Dieses anspruchsvolle aber leicht erlernbare Textverarbeitungsprogramm für den C 64 kommt aus den USA. Besonders zu bewerten ist bei diesem Programm die einfache und übersichtliche Bedienung, die unter anderem durch einzelne Menüs mit Bildschirmsymbolen erzielt wird. Inhaltlich verfügt dieses Textverarbeitungsprogramm unter anderem über Such- und Korrekturvorgänge, Formatierungsvorgänge als auch Seitennumerierung. Die Lizenz für die deutschsprachige Version hat der **Langenscheidt-Verlag,** der dieses Programm für ca. 130 DM anbietet.

Der „Neue von TOSHIBA“: T1500

IBM-kompatibler Personalcomputer mit anerkannt hohem Leistungsstandard

Professionelle Personalcomputer zeichnen sich dadurch aus, daß sie die Fähigkeit besitzen, problemorientierte Arbeitsgänge in den verschiedensten Anwendungsbereichen zu erleichtern, zu beschleunigen und letztendlich damit auch die Kosten zu senken. Der japanische Hersteller TOSHIBA, weltweit ein starker Partner in der Informationsverarbeitung, hat deshalb auch in Deutschland bei allen Produktneuheiten größten Wert auf diese Grundforderung der Anwender gelegt. Das abgerundete Software-Angebot tat ein übriges, um die Wettbewerbsvorteile der TOSHIBA-Produkte deutlich in den Vordergrund zu stellen.

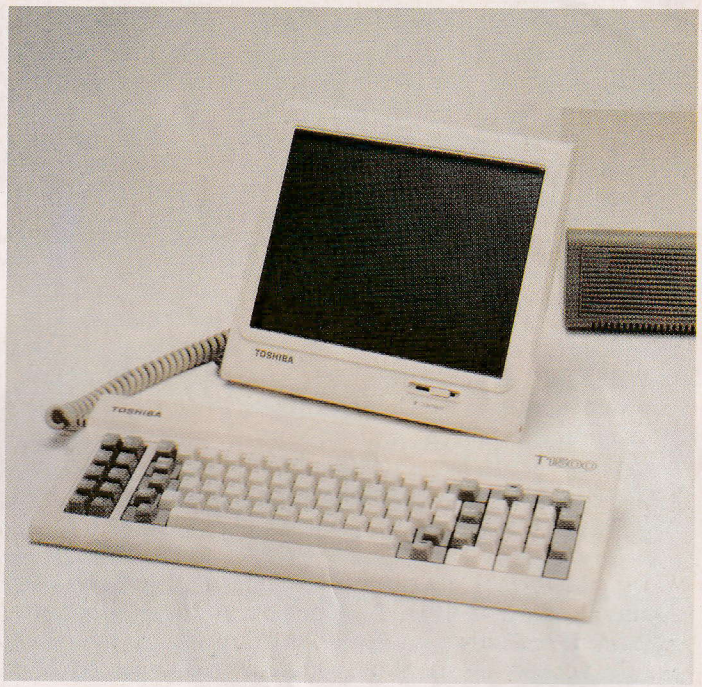
Auf der MICRO-COMPUTER '85 hat TOSHIBA jetzt erstmals eine zukunftsweisende Variante seines Angebotes vorgestellt: Micro-Computer TOSHIBA T1500. Als IBM/PC-kompatibles Gerät kann der T1500 jetzt das gesamte Spektrum der auf dem Markt befindlichen PC-Anwendungssoftware und alle Systemerweiterungen nutzen. Und zwar mit einer Hardware, die TOSHIBA's umfangreiche Erfahrungen auf den Weltmärkten widerspiegelt.

In der Grundausstattung ist der TOSHIBA T1500 mit dem leistungsstarken Processor Intel 8088 ausgestattet. Die Taktfrequenz beträgt dabei 4.77 MHz. Mit 128 KB Systemspeicher, Color/Grafik-Adapter, Anschluß für LCD-Bildschirm, 2 slim-line Diskettenlaufwerken mit je 320/360 KB Kapazität, 3 PC-kompatiblen Erweiterungs-Steckplätzen und MS-DOS kann sich der „Neue von TOSHIBA“ im Reigen des Mitbewerbs durchaus sehen lassen.

Um die Einsatzmöglichkeiten noch zu erweitern, bietet TOSHIBA darüber hinaus auch einen grünen Bildschirm, Farbbildschirm, flachen LCD-Bildschirm mit 25 Zeilen x 80 Zeichen. Kommunikations-Adapter, Speichererweiterung bis 640 KB auf der Mutterplatine, Festplattenlaufwerk (10 MB) und 8087 Co-Processor.

Mit diesen Eigenschaften dürfte der TOSHIBA T1500 zu den leistungsfähigsten und von seinen Einsatzmöglichkeiten her flexibelsten IBM-kompatiblen Personalcomputern zählen.

Nähere Informationen durch: TOSHIBA Europa (I.E.) GmbH, IPS-Division, Hammer Landstr. 115, 4040 Neuss 1, Tel.: 0 21 01 / 158-0.



TOSHIBA T1100: der Personal-Computer in Aktenkoffer-Format

TOSHIBA hat auf der MICRO-COMPUTER '85, die im Januar/Februar 1985 auf dem Frankfurter Messegelände veranstaltet wurde, auch einen Personal-Computer im Aktenkoffer-Format vorgestellt: T1100, ein Rechner, der mit seinem 16-bit-Prozessor 80C88, 256 KB Arbeitsspeicher und 720 KB Diskettenlaufwerk keine Wünsche professioneller Benutzer offen läßt und das, obwohl er nur 4,1 kg wiegt.

Auf dem Flüssigkristall-Bildschirm sind bis zu 80 Zeichen in 25 Zeilen darstellbar, außerdem läßt er sich auf jeden Blickwinkel (0 bis 180 Grad) durch vertikale Verstellung optimal anpassen. Selbst anspruchsvolle Grafiken sind darstellbar, denn die Auflösung von 640 x 200 oder 320 x 200 garantiert eine ausreichende Bildqualität.

Um die Leistungsfähigkeit und die Anwendungsbreite dieses „Leichtgewichtes“ noch zu erhöhen, besteht die Möglichkeit der Erweiterung von zusätzlich 256 KB Arbeitsspeicher, ein zweites, externes Diskettenlaufwerk, Drucker, Farbbildschirm und die RS-232-Schnittstelle.

Das Besondere am „Aktenkoffer-Computer“ T1100 ist – neben den Leistungsmerkmalen – der flexible Einsatz sowohl unterwegs als auch im Büro: Unterwegs wird er über eingebaute Batterien mit Strom versorgt, im Büro kann man dann auf eine externe Stromversorgung zurückgreifen und das Gerät an einen vorhandenen Farbbildschirm anschließen .

Trotz seiner geringen äußeren Abmessungen, ist auch der T1100 vollständig kompatibel zum IBM-PC. Diese Eigenschaft versetzt ihn in die Lage, das gesamte bewährte Software-Angebot des Weltmarktes einsetzen zu können.

Nähere Informationen durch: TOSHIBA Europa (I.E.) GmbH, IPS-Division, Hammer Landstr. 115, 4040 Neuss 1, Tel.: 0 21 01 / 158-0.

Wörterbuch

Fortsetzung Wörterbuch

A →→ A →→

Ablaufdiagramm:
(Flußdiagramm-flow chart) In einem Ablaufdiagramm werden die einzelnen Operationsschritte mit Hilfe von Standard-Symbolen grafisch dargestellt. Die wichtigsten Ablaufdiagramme sind: Programmablauf-, Datenflußplan- und Befehlsdiagramm.

Adreßbus:
Speziell für die Übertragung von Adressen vorgesehene Datenleitungen aus mehreren Kanälen, die zur Kommunikation zwischen verschiedenen Komponenten eines Computersystems verwendet werden.

Akkumulator:
Eines der wichtigsten Register in einem Mikroprozessor, da hier die Zwischenergebnisse aller Operationen und Funktionen gespeichert werden. Es werden also alle entsprechenden Zwischenergebnisse in einen externen Speicher geschrieben, der, wenn das Ergebnis feststeht, wieder gelöscht wird.

Akustikkoppler:
(Acoustik coupler) Bei einem Terminalgerät die Anschlußeinheit für die Telefon-Sprechmuschel um Datenkommunikation über längere Strecken zu ermöglichen.

Assemblieren:
Ein in Assemblersprache geschriebenes Programm wird in den Maschinencode sowie die Maschinenbefehle umgewandelt.

ALGOL:
Eine höhere Programmiersprache für den technisch-wissenschaftlichen Bereich (Algorithmic Language).

ALU:
(Arithmetic logic unit) Ist derjenige Teil des Microprozessors, der für das Rechnen zuständig ist. Wie der Begriff bereits aussagt, können hier arithmetische und logische Operationen wie Addition von Binärzahlen, Boolesche Operationen etc. durchgeführt werden.

Array:
Ist der Bereich von Feldern, die gleiche Eigenschaften haben und durchnummeriert sind.

ASCII:
(American standard code for information interchange) Von Computer-Insidern wird dieser Code immer als ASCII Code bezeichnet. Dieser Code legt genau fest, wie Buchstaben, Zahlen, graphische Zeichen etc. digital dargestellt werden, so daß sie ein Computer verstehen und bearbeiten kann.

B →→ B →→

BCD:
(Binary coded decimal) Ist der am häufigsten verwendete Computercode (binär codiertes Dezimalsystem). Da Computer nur mit zwei Zahlen, 0 oder 1 (Strom oder kein Strom) arbeiten, ist die Möglichkeit geschaffen worden, eine Dezimalzahl genau mit diesen beiden Zuständen zu kennzeichnen, der BCD-Code. Dieser Code besorgt das mit jeweils vier Bit pro Ziffer des Dezimalsystems.

Baud (bd):
Dieser Begriff, der aus der Nachrichtentechnik stammt, ist die Maßeinheit für die Schrittgeschwindigkeit bei der Datenübertragung und gibt an, wie viele Zeichen pro Sekunde vom Computer zu einem anderen oder zu einem Drukker übertragen werden. Jedoch muß genau zwischen Schrittgeschwindigkeit und Übertragungsgeschwindigkeit unterschieden werden, denn nur, wenn pro Schritt auch ein Binärzeichen, also ein Bit übertragen wird, ist die Schrittgeschwindigkeit auch gleich der Übertragungsgeschwindigkeit. Letztere wird in Bit pro Sekunde angegeben (BpS).

Basic:
(Beginners all purpose symbolic instructions code) Eine der beliebtesten (wenn auch langsamen) und am weitesten verbreitete Programmiersprache bei Home- als auch Personalcomputern.

Bidirektionaler Drucker:
(Bidirectional Printer) Bei diesen Drukkern werden die Zeichen in beiden Richtungen ausgedruckt. Somit entfällt die Wagenrücklaufzeit, was wiederum eine höhere Schreibgeschwindigkeit des Druckers zur Folge hat.

Bit:
(Binärzeichen) Ein Bit ist die kleinste Einheit des Dualsystems. Ein Bit kann sowohl den Wert 0 als auch den Wert 1 annehmen. Jede Speicherstelle eines Computers kann genau ein Bit aufnehmen. Durch die unterschiedliche Aneinanderreihung mehrerer Bits ergeben sich Informationen, die der Computer versteht und verarbeitet. Hierbei werden jeweils 8 Bit zu einem Byte zusammengefaßt.

MSX-Zeichensatz-Konverter

Neu in CPU: Programme für MSX

MSX-Computer sind mit einer parallelen Schnittstelle ausgestattet und können jeden Drucker mit Parallelschnittstelle steuern.
Viele Drucker können jedoch nicht den kompletten 254 Zeichen umfassenden MSX-Zeichensatz darstellen, so daß der Benutzer gezwungen ist, einen speziellen MSX-Drucker zu kaufen, auch wenn bereits ein Drucker vorhanden ist.
Der MSX-Zeichensatz-Konverter schafft dem Abhilfe, denn mit Hilfe dieses Programmes kann der komplette MSX-Zeichensatz auf jedem grafikfähigen Nadeldrucker ausgedruckt werden. Das Programm liest den kompletten MSX-Zeichensatz aus dem Videoram, wandelt diesen in für den Drucker verständliche Grafik um und speichert diesen umgewandelten Zeichensatz in einem Textvariablenfeld ab.
Dann werden die auszudruckenden Dateien von Cassette gelesen und abgearbeitet. An den Drucker wird jetzt nicht mehr der ASCII-Code des Zeichens übertragen, sondern die umgewandelte Grafik des entsprechenden Zeichens. Somit kann der Drucker auch die Grafikzeichen ausgeben.
Das Programm berücksichtigt auch die Grafikzeichen, die durch 2 Bytes dargestellt werden. Somit werden alle 254 Zeichen für den Drucker darstellbar.
Das Programm ist mit einem SEIKOSHA-Drucker geschrieben und ausgetestet. Da jetzt verschiedene Drucker die Grafik anders ansprechen, muß gegebenenfalls die Grafik-Steuer-Variable in Zeile 260 angepaßt werden. Dies ist jedoch die einzige Änderung, die vorkommen kann.
Zeile 260 lautet wie folgt:
260 P = CHR$ (27) + "K008"
Hierin bedeutet: CHR$ (27) ist die ESC-Sequenz, die dem Drucker mitteilt, daß eine Sonderfunktion eingestellt wird. "K008" bedeutet, daß die nächsten 8 Zeichen als Grafik interpretiert werden sollen.

Doch nun zum Programmablauf:
MSX-Zeichensatz-Konverter können sämtliche im ASCII-Format gespeicherten Dateien von Cassette ausgeben. Um eine Datei ausgeben zu lassen, muß diese zunächst auf Cassette gespeichert werden. Dies geht im Normalfall vom Programm her mit einem OPEN-Befehl und anschließend mit PRINT #.
Um ein Listing mit richtigem MSX-Zeichensatz auszudrucken, muß das Programm mit SAVE "CAS:Name" abgespeichert werden, wobei Name jeder beliebige Programmname sein darf, da der Zeichensatz-Konverter den Programmnamen nicht abfragt.
Nachdem Sie Ihre Datei (oder Programm) auf Cassette gespeichert haben, laden Sie den Zeichensatz-Konverter ein und starten diesen mittels der Funktionstaste F5.
Auf dem Bildschirm sehen Sie jetzt den Schriftzug:

Zeichensatz wird umgewandelt bitte warten
Darunter erscheint die Schrift : Zeichen-> ### und Sie sehen aufwärtszählende Nummern. Dies gibt an, welches Zeichen gerade umgewandelt wird. Diese Umwandlung dauert ca. 4 Minuten, und sobald Zeichen 255 umgewandelt ist, werden Sie aufgefordert, das Band zurückzuspulen, wobei der Recordermotor eingeschaltet wird. Wenn das Band zurückgespult ist, drücken Sie die <RETURN>-Taste und Sie werden aufgefordert, die Wiedergabe zu starten.
Sobald der Rechner jetzt eine ASCII-Datei gefunden hat, gibt er die Meldung „Druckvorgang ist eingeleitet" aus. Die Datei wird jetzt Zeile für Zeile eingelesen, abgearbeitet und an den Drucker ausgegeben, wobei die zur Zeit aktuelle Zeile der Datei im unteren Drittel des Bildschirmes ausgegeben wird.
Nachdem die Datei komplett ausgedruckt wurde, erscheint die Frage nach weiteren Ausdrucken. Wenn Sie noch weitere Dateien ausdrucken wollen, so antworten Sie mit „J", und es geht weiter mit der Aufforderung, das Band zurückzuspulen. Somit sparen Sie sich die Zeit der Zeichensatzumwandlung.
Wenn Sie keine Dateien mehr auszudrucken haben, so antworten Sie mit „N", um das Programm zu beenden.

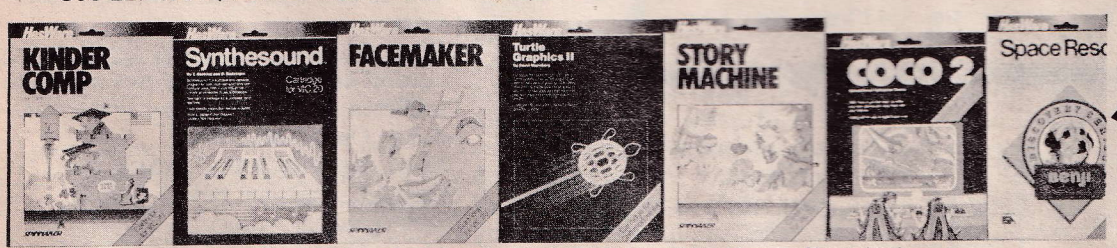
```
100 '●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
110 '●                            ●
120 '●           M S X            ●
130 '●    Zeichensatz Konverter   ●
140 '●                            ●
150 '●»«»«»«»«»«»«»«»«»«»«»«»«»«»«»●
160 '●                            ●
170 '●  (c) 1985 by Volker Becker ●
180 '●                            ●
190 '●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
200 '
210 '
220 CLEAR 25000
230 DEFINT I,J,K,L
240 DEFSTR P,X,Y,Z
250 DIM Y(8),Z(255)
260 P=CHR$(27)+"K008"
270 KEY OFF:CLS:SCREEN 0:COLOR 14,1,1
```

Vorsicht

Wenn Sie den Zeichensatz-Konverter mittels <CTR>+<STOP> abbrechen, so muß unbedingt der Befehl CLEAR 200 gegeben werden, da der Zeichensatz-Konverter einen Bereich von 25 000 Bytes für Textfelder reserviert.
Wird CLEAR 200 nicht gegeben, so kommt es beim Einlesen des nächsten größeren Programmes zur Fehlermeldung „Out of Memory".
Wird das Programm jedoch mit „N" beendet, muß der Befehl CLEAR 200 nicht gegeben werden.
Die Speicherbelegung von 25 000 Bytes bleibt jedoch nur so lange bestehen, bis die <RESET>-Taste gedrückt wird.

```
280 LOCATE 6,0:PRINT "MSX Zeichensatz Konverter"
290 LOCATE 6,1:PRINT "--------------------------"
300 LOCATE 6,3:PRINT "(c) 1985 by Volker Becker"
310 LOCATE 0,5:PRINT STRING$ (37,195)
320 LOCATE 7,7:PRINT "Zeichensatz umwandeln"
330 LOCATE 11,8:PRINT "bitte  warten"
340 LOCATE 11,10:PRINT "Zeichen->"
350 LOCATE 0,12:PRINT STRING$ (37,195)
360 FOR I=0 TO 255:LOCATE 21,10:PRINT USING "###";I:FOR K=0 TO 7:Y(K+1)=BIN$(VPE
EK(&H800+8*I+K))
370 IF LEN(Y(K+1))=8 THEN 390
380 FOR J=1+LEN(Y(K+1)) TO 8:Y(K+1)="0"+Y(K+1):NEXT
390 NEXT
400 FOR K=1 TO 8:X="":FOR J=8 TO 1 STEP-1:X=X+MID$(Y(J),K,1):NEXT J:Z(I)=Z(I)+CH
R$(VAL("&B"+X)):NEXT K,I
410 RESTORE 730:FOR I=1 TO 7:READ L:Z(L)="":FOR K=0 TO 7:READ J:Z(L)=Z(L)+CHR$(J
):NEXT K,I
420 FOR I=6 TO 11:LOCATE 0,I:PRINT STRING$(37,32):NEXT
430 LOCATE 9,7:PRINT "Spule Band zurück"
440 LOCATE 19,8:PRINT "┌──────┐"
450 LOCATE 8,9:PRINT "Dann drücke|RETURN|"
460 LOCATE 19,10:PRINT "└──────┘":BEEP:MOTOR ON
470 IF INKEY$=CHR$(13) THEN 480 ELSE 470
480 MOTOR OFF
490 LOCATE 9,7:PRINT "Starte Wiedergabe":BEEP
500 IF INKEY$=CHR$(13) THEN 510 ELSE 500
510 FOR I=6 TO 11:LOCATE 0,I:PRINT STRING$(37,32):NEXT
520 LOCATE 11,7
530 OPEN "CAS:" FOR INPUT AS #1
540 LOCATE 4,9:PRINT "Druckvorgang ist eingeleitet"
550 IF EOF(1) THEN 630
560 LINE INPUT #1,X
570 FOR I=14 TO 21:LOCATE 0,I:PRINT STRING$(37,32):NEXT
580 LOCATE 0,14:PRINT X
590 FOR I=1 TO LEN(X):J=ASC(MID$(X,I,1)):IF J>31 THEN 610
600 I=I+1:J=ASC(MID$(X,I,1))-64
610 LPRINT P;Z(J);:NEXT:LPRINT
620 GOTO 550
630 FOR I=14 TO 21:LOCATE 0,I:PRINT STRING$(37,32):NEXT
640 CLOSE
650 FOR I=6 TO 11:LOCATE 0,I:PRINT STRING$(37,32):NEXT
660 LOCATE 8,7:PRINT "Weitere Ausdrucke ?"
670 LOCATE 15,9:PRINT "[J/N]"
680 X=INKEY$
690 IF X="j" OR X="J" THEN 420
700 IF X="n" OR X="N" THEN 720
710 GOTO 680
720 CLEAR 200:CLS:COLOR 14,4,4:END
730 DATA 63,2,1,1,89,6,0,0,0,70,127,17,17,17,1,0,0,0,80,127,17,17,17,14,0,0,0,82
,127,17,17,49,78,0,0,0,102,0,16,126,17,2,0,0,0,158,255,17,14,8,253,136,0,0,159,1
28,144,116,17,1,0,0,0
```

Human Engineered Software, 150 North Hill Drive, Brisbane, CA 94005
800-227-6703 (in California 800-632-7979) Dept. C20



HesWare™

HesWare is a trademark of Human Engineered Software. Facemaker is a trademark of Spinnaker Software. VIC 20 and Commodore 64 are trademarks of Commodore Electronics Ltd. Atari is a registered trademark of Atari, Inc. IBM is a registered trademark of International Business Machines.

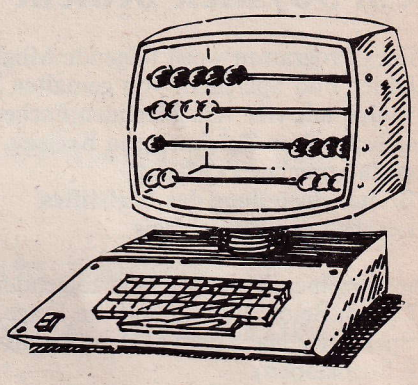
© 1983

CIRCLE 133 ON READER SERVICE CARD

RAM-Tester für den Atari

Ein Programm, das vom Basic aus den RAM-Speicher testet

Mit Hilfe dieses Programmes können alle Besitzer des alten Atari 800, der noch nicht mit dem Speichertest des neuen Atari 800 XL ausgestattet ist, vom Basic aus den RAM-Speicher testen.
Nach dem Start wird der testbare Speicherbereich angegeben.
Nun grenzen Sie den zu testenden Bereich mit >Anfang Testbereich< und >Ende Testbereich< ein, worauf dieser angegebene Bereich überprüft wird. Dies geschieht durch ein kleines Maschinenprogramm, das in Zeile 2900 eingelesen wird.
Ist die Überprüfung ordnungsgemäß verlaufen, so meldet sich Ihr Computer mit *** Alles in Ordnung ***. Stellt der Computer einen Fehler fest, so wird dieser angegegeben. Um fortzufahren, drücken Sie eine beliebige Taste.



```
10 REM ********************************
20 REM *                              *
30 REM *        RAM Tester            *
40 REM *    Fritz NiesPodziani        *
50 REM *                              *
60 REM ********************************
100 N1=1:N2=N1+N1:N255=255:N256=N255+N1
200 DIM S$(N2):S$(N1,N1)=CHR$(157):S$(N2,N2)=CHR$(159)
298 REM MaschienensPrache Programm
300 GOSUB 2900
400 L=PEEK(15)*N256:H=PEEK(742)*N256:IF PEEK(14)<>N0 THEN L=L+N256
500 ? CHR$(125);S$;"Atari SPeichertester";CHR$(155);S$;"(c) Fritz NiesPodziani,1
984"
510 ? :?
550 ? "SPeicherbereich :"
600 ? S$;"Untere Grenze:";L:? S$;"Obere Grenze:";H
700 ? S$;"Geben Sie den Bereich an";CHR$(155)
800 TRAP 800:? S$;"Anfang Testbereich:";:INPUT LOW:IF LOW<L OR LOW>H THEN 800
900 TRAP 900:? S$;"Ende Testbereich:";:INPUT HIGH:IF HIGH>H OR HIGH<L OR HIGH-LO
W<N256 THEN 900
1000 POKE 205,N0:POKE 206,INT(HIGH/N256)
1100 TRAP 32767:POKE 203,N0:POKE 204,INT(LOW/N256)
1200 POKE 764,N255
1300 POKE 764,N255:X=USR(1536)
1400 IF PEEK(208)=N0 THEN 2200
1500 ? " Fehler in ";(PEEK(203)+PEEK(204)*N256);" EXP=";PEEK(207);" ACT=";PEEK(2
09)
1600 SOUND N0,PASS,6,8:FOR I=N1 TO 5:NEXT I:SOUND N0,N0,N0,N0
1699 REM
1700 IF PEEK(203)=N255 THEN POKE 204,(PEEK(204)+N1):POKE 203,N0:GOTO 1900
1800 POKE 203,(PEEK(203)+N1)
1900 POKE 764,N255:POKE 559,34
2000 IF PEEK(764)=N255 THEN 2000
2100 GOTO 1300
2200 ? "*** Alles in Ordnung ***  "
2210 END
2300 FOR I=N1 TO 5:NEXT I:SOUND N0,N0,N0,N0
2400 IF PEEK(764)<>N255 THEN 2600
2500 GOTO 1100
2600 POKE 764,N255
2700 POKE 559,34:IF PEEK(764)=N255 THEN 2700
2800 GOTO 1100
2900 FOR L=1536 TO 1576:READ H:POKE L,H:NEXT L:RETURN
3000 DATA 104,169,0,160,0,24,145,203,209,203,208,18,105,1,208,246,200,208,242,23
0,204,166,204,228,206
3100 DATA 208,234,133,208,96,133,207,177,203,133,209,169,1,133,208,96
```

GNOM-PAINTER für Atari

GNOM-PAINTER ist ein Malprogramm für den Atari 800, das mit dem Joystick bedient wird.

Dieses Programm weist folgende Möglichkeiten auf:
- Laden und Speichern von gemalten Bildern
- **Malen mit vier verschiedenen Farben,** die einzelnen in Farbstufe und Helligkeit veränderbar
- **selbständiges Zeichnen von Kreisen, Ellipsen**
- **symmetrisches Zeichnen**
- **Breiteneinstellung des Malstiftes**
- **Löschen des Bildschirms**

Die einzelnen Vorgänge sind folgendermaßen durchführbar:
- **Laden:** Beim Titelbild auf die Frage mit „J" antworten, dann den zu ladenden Namen eingeben.
- **Speichern:** Nach dem Malen „Return" drücken und den Namen, unter dem abgespeichert werden soll, eingeben (höchstens 8 Buchstaben).
- **Farbenwechsel:** „F" drücken. Am rechten unteren Bildschirmrand erscheint die gerade benutzte Farbe.
- **Farbenänderung:** „A" drücken. Zur Farbänderung Joystick nach links oder rechts bewegen. Zur Helligkeitsänderung Joystick nach oben oder unten bewegen. Nachdem die gewünschte Farbe auf dem Screen erscheint, Feuerknopf drücken.
- **Kreis:** „C" drücken. Danach 1. Zahl eingeben (Radius), dann 2. Zahl eingeben (0 oder 1). 0 bedeutet, daß der Kreis ausgefüllt ist, 1 das der Kreis nur umrandet ist.
- **Ellipse:** „E" drücken. Danach 1. Zahl eingeben (Breite horizontal), dann 2. Zahl (Breite vertikal) eingeben, 3. Zahl eingeben (siehe 2. Zahl beim Kreis). Soll der Kreis oder die Ellipse vorzeitig abgebrochen werden, so müssen Sie den Feuerknopf drücken.
- **symmetrisches Zeichnen:** „S" drücken.
- **verbreitern des Stiftes:** „B" drücken
- **Löschen:** „L" drücken.

Anmerkung

Der Strich in Zeile 112 wird durch „Control" und „M" erzeugt.

```
4 GOSUB 110
5 OPEN #1,4,0,"K:":DEG :DIM N$(16),P(10)
6 INPUT N$:IF N$(1,1)="J" THEN GOTO 80
9 N$="D:":GRAPHICS 23:R=1:COLOR R:PLOT 159,90:DRAWTO 159,95:DRAWTO 154,95
10 R=1:GOSUB 100
11 S=STICK(0):X=X+(S=7)-(S=11)+(S=6)-(S=10)+(S=5)-(S=9):Y=Y+(S=13)-(S=14)+(S=5)+
(S=9)-(S=10)-(S=6):GOSUB 60
12 IF PEEK(764)=255 THEN GOTO 24
13 GET #1,A:IF A=83 THEN SY=SY+1:IF SY>1 THEN SY=0
14 IF A=67 THEN GET #1,A1:GET #1,A2:A1=2*(A1-48):A2=A2-48:A3=A1:GOTO 50
15 IF A=69 THEN GET #1,A1:GET #1,A3:GET #1,A2:A1=2*(A1-48):A2=A2-48:A3=2*(A3-48)
:GOTO 50
16 IF A=70 THEN R=R+1:R=R-4*(R)4):COLOR R:PLOT 159,90:DRAWTO 159,95:DRAWTO 154,9
5
17 IF A=65 THEN R=R+(R=4):GOSUB 65
18 IF A=155 THEN GOTO 70
19 IF A=76 THEN ? #6;")"
20 IF A=66 THEN BRE=BRE+1:IF BRE=2 THEN BRE=0
24 POKE 53248,X+44:IF S<>15 AND S<>11 AND S<>7 THEN FOR J=1 TO 7:POKE PM+511+Y+J
+14,P(J):NEXT J
25 IF STRIG(0)=0 THEN COLOR R:PLOT X,Y:IF SY=1 THEN PLOT 158-X,Y:PLOT 158-X,94-Y
:PLOT X,94-Y
```

```
26 IF STRIG(0)=0 AND BRE=1 AND X>0 AND X<158 AND Y<94 AND Y>0 THEN GOSUB 35
30 GOTO 11
35 COLOR R:PLOT X-1,Y-1:DRAWTO X+1,Y-1:DRAWTO X+1,Y+1:DRAWTO X-1,Y+1:DRAWTO X-1,
Y-1:RETURN
40 FOR T=0 TO 20:POKE 53279,80:NEXT T:RETURN
50 FOR T=0 TO 360 STEP 2:A4=COS(T)*A1+X:A5=SIN(T)*A3+Y:IF A4<0 OR A4>159 OR A5<0
 OR A5>95 THEN 54
51 COLOR R:PLOT A4,A5:IF A2=1 THEN DRAWTO X,Y
52 IF SY=1 THEN PLOT 158-A4,A5:PLOT 158-A4,94-A5:PLOT A4,94-A5
53 IF A2=1 AND SY=1 THEN PLOT 158-A4,A5:DRAWTO 158-X,Y:PLOT 158-A4,94-A5:DRAWTO
158-X,94-Y:GOSUB 56
54 IF STRIG(0)=0 THEN GOSUB 40:GOTO 25
55 NEXT T:SOUND 0,10,10,10:SOUND 0,0,0,0:GOSUB 40:GOTO 25
56 PLOT A4,94-A5:DRAWTO X,94-Y:RETURN
60 IF X>158 THEN X=158
61 IF X<0 THEN X=0
62 IF Y<0 THEN Y=0
63 IF Y>94 THEN Y=94
64 RETURN
65 SETCOLOR R-1,D,E:S=STICK(0):D=D+(S=7)-(S=11):D=D-15*(D>15)+15*(D<0):E=E+(S=13
)-(S=14):E=E-15*(E>15)+15*(E<0)
66 IF STRIG(0)=0 THEN GOSUB 40:RETURN
67 GOTO 65
70 P=PEEK(88)+256*PEEK(89):FOR T=3 TO 14:GET #1,A:IF A=155 THEN GOTO 72
71 N$(T,T)=CHR$(A):NEXT T
72 OPEN #2,8,0,N$:FOR T=0 TO 3800:IF PEEK(P+T)<>0 THEN PUT #2,INT(T/40):PUT #2,T
-INT(T/40)*40:PUT #2,PEEK(P+T)
73 NEXT T:CLOSE #2:GOTO 11
80 POSITION 20,22:? "UNTER WELCHEM NAMEN";:INPUT N$:FOR T=12 TO 1 STEP -1:N$(T+2
,T+2)=N$(T,T):NEXT T
81 N$(1,1)="D":N$(2,2)=":"
82 GRAPHICS 23:P=PEEK(88)+256*PEEK(89):R=1:X=0:Y=0
83 OPEN #2,4,0,N$:TRAP 86
84 FOR T=0 TO 4:GET #2,A:POKE 708+T,A:NEXT T:FOR I=0 TO 3800:GET #2,T:GET #2,T1:
GET #2,A
85 POKE P+T*40+T1,A:NEXT I
86 CLOSE #2:GOTO 10
100 A=PEEK(106)-32:POKE 54279,A:PM=A*256:POKE 559,46:POKE 53277,3:POKE 53248,X+4
4
101 FOR J=PM+512 TO PM+640:POKE J,0:NEXT J:POKE 704,14
102 FOR J=1 TO 7:READ A:POKE PM+511+14+J,A:P(J)=A:NEXT J:DATA 0,20,0,20,0,0,0
103 RETURN
110 GRAPHICS 0:POKE 82,0:POKE 709,25:POKE 710,25:P=PEEK(106)-8:PM=P*256:POKE 756
,P:POKE 752,1
111 POKE 712,25:POKE 708,25:DL=PEEK(560)+256*PEEK(561):POKE DL+6,7:POSITION 0,1:
? "    GNOM-PAINTER"
112 POSITION 28,1:? "──────────────────────────────":POSITION 32,0:? "R"
124 POSITION 20,19:? "Willst du ein Bild einladen";
125 FOR T=PM TO PM+256:POKE T,0:POKE T+256,0:POKE T+512,0:POKE T+768,0:NEXT T:PO
KE 709,0:POKE 708,0
126 PO=57344:FOR I=31*8 TO 58*8 STEP 8
127 FOR T=0 TO 7:POKE I+T+PM,PEEK(PO+I+T):POKE T+I+PM+512,PEEK(PO+I+T+512):SOUND
 0,T*30+10,8,10:NEXT T:NEXT I
128 FOR T=104 TO 111:POKE T+PM,PEEK(57344+T):POKE T+PM+512,PEEK(57344+512+T):SOU
ND 0,(T-104)*30+10,8,10:NEXT T
129 SOUND 0,0,0,0:RETURN
```

Cherry Harry für ATARI 600 XL

Gärtner Harrys gefährliche Ernte

Gärtner Harry entdeckt in seinem Garten riesige Kirschen, die aber nur im Erdreich und nicht etwa an Bäumen wachsen. Ein guter Gärtner bringt natürlich ertragreiche Ernten ein, also gräbt sich Harry durch das Erdreich, um an die Früchte seiner Arbeit zu gelangen.
Diesem Unterfangen sind natürlich einige gefahrvolle Hindernisse in den Weg gestellt. Harry wird von feindlichen Erdwesen verfolgt, die von ihren Schußwaffen liebend gern Gebrauch machen, zudem behindern ihn Steine bei seinen Erdarbeiten und ab Runde 2 bedroht ihn auch noch eine Geisterkette.
Der wagemutige Gärtner muß in jeder Runde 30 Kirschen ausgraben. Doch die Erdwesen umgeben die Kirschen, nachdem Harry bereits 5 Exemplare eingesammelt hat, mit einem Schutzmantel. Dieser Schutzmantel kann nur zerstört werden, wenn Harry die Erdwesen per „Pumprohr" aufpumpt und ihrem Dasein dadurch ein Ende setzt. Nach ihrem „Dahinscheiden" verwandeln sich die Erdwesen in Steine, erhalten jedoch nach kurzer Zeit ihre ursprüngliche Gestalt zurück.

Zerstörung der Erdwesen:
Wenn man den Joystick nach oben drückt, wirft „Cherry Harry" ein Pumprohr nach oben aus, entsprechend umgekehrt. Wenn es gelungen ist, ein Erdwesen mit dem Pumprohr zu berühren, müssen folgende Aktionen durchgeführt werden, um es endgültig zu eliminieren:
1. Bleiben Sie auf dem betätigten Feuerknopf, bis das entstandene Geräusch verstummt ist.
2. Drücken Sie sofort wieder und so lange, bis das neu entstandene Geräusch verstummt ist.
3. Führen Sie Schritt 2 noch zweimal aus, dann platzt das Erdwesen und der Schutzmantel verschwindet.
Während des Pumpvorganges bleibt das Erdwesen geschockt stehen, wenn Sie aber den Feuerknopf zu früh oder zu spät loslassen oder nicht noch einmal drücken, bewegt es sich wieder.

Screenanzeige:
Außer Runde und Punktzahl werden auch die Ersatzleben, von denen Harry anfangs drei besitzt, rechts oben angezeigt. Darunter, mit einer Kirsche als Symbol, sind die noch aufzusammelnden Kirschen aufgeführt. Wenn dort eine Null steht, muß das Erdwesen aufgeblasen werden, erst dann kann man weitersammeln.

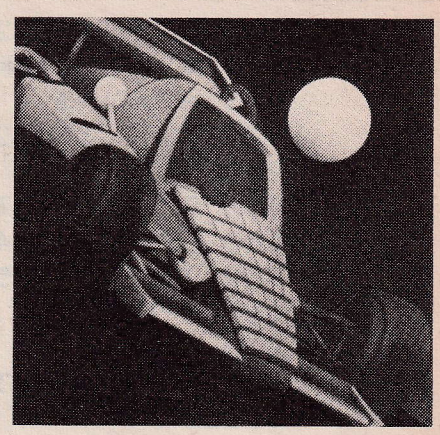
Gesteuert wird mit Joystick

```
0 REM ********************************
1 REM *         CHERRY HARRY         *
2 REM ********************************
3 REM * (c) 1985 by Markus Loeffler *
4 REM ********************************
5 REM *   Computer: ATARI 600/800    *
6 REM *   Joystick erforderlich!     *
7 REM ********************************
8 REM
9 REM
10 DIM T$(6):L=0
13 GOSUB 1000:POKE 559,L:A=14336:RESTORE 10000
14 REM *** INITIALISIERUNG ***
15 READ T:IF T=-1 THEN 30
20 FOR I=A+T*8 TO A+T*8+7:READ C
25 POKE I,C:NEXT I:GOTO 15
30 B=11328:POKE 54279,40:POKE 53277,3:POKE 623,1:POKE 704,172:POKE 705,172
35 FOR H=L TO 1:B1=B+256*H+H*8:FOR I=B1 TO B1+145 STEP 16:RESTORE 10200
40 FOR U=I TO I+7:READ C:POKE U,C:NEXT U:NEXT I:NEXT H
60 FOR I=128 TO 440:SOUND L,I,10,10:POKE A+I,PEEK(57344+I):SOUND L,L,L,L:NEXT I
90 PN=1:RN=1:LB=3:LZ=53247:T=500
95 REM *** SPIELFELDAUFBAU ***
100 GRAPHICS 17:POKE 756,A/256:POKE 559,L
105 SETCOLOR L,1,6:SETCOLOR 4,L,L:SETCOLOR 1,1,10:SETCOLOR 2,3,8:SETCOLOR 3,7,10
110 COLOR 34:FOR I=5 TO 23
115 PLOT L,I:DRAWTO 19,I:NEXT I
130 POSITION 2,1:? #6;"runde:";RN:POSITION 14,1:? #6;CHR$(6);":";LB
135 POSITION 2,2:? #6;"punkte:";PN:POSITION 14,2:? #6;CHR$(3);":5"
140 COLOR 1:PLOT L,L:DRAWTO L,23:DRAWTO 19,23:DRAWTO 19,L:DRAWTO L,L:COLOR 41:PL
OT 1,4:DRAWTO 18,4
145 COLOR 43:FOR I=1 TO RN*7:PLOT RND(L)*17+1,RND(L)*16+6:NEXT I
150 COLOR 5:PLOT 10,10:COLOR 163:FOR I=1 TO 25
```

```
160 M=INT(RND(L)*17+1):N=INT(RND(L)*16+6):LOCATE M,N,G:IF G<>34 OR (M=10 AND N=1
0) THEN 160
170 SOUND L,U,10,10:SOUND L,L,L,L:PLOT M,N:NEXT I
180 POKE 559,62:POKE 53248,0:POKE 53249,0
185 Z=L:ZA=L:PP=1:QQ=L:LL=5:MH=0:P=10:Q=10
190 X=10:Y=10:UI=10:XE=32:XP=53248:WT=12-RN*2:COLOR 5:PLOT 10,10:IF RN=1 THEN WT
=30
195 IF STICK(L)=15 THEN 195
197 REM *** HAUPTSCHLEIFE ***
200 FOR I=L TO WT:NEXT I:Z=Z+1:IF Z=40-RN*4 THEN GOSUB 900
210 S=STICK(L):XX=L:YY=L:T=T-1:IF S=15 THEN 300
220 XX=(S=7)-(S=11):YY=(S=13)-(S=14):IF XX<>L THEN TT=XX
225 IF YY<>L THEN KK=YY
230 LOCATE X+XX,Y+YY,G:IF G=32 THEN 290
240 IF G=43 THEN 5000
250 IF G=163 AND LL>L THEN FOR I=255 TO 0 STEP -15:SOUND L,I,10,10:NEXT I:GOSUB
970:LL=LL-1:GOSUB 950:GOTO 290
260 IF G=34 THEN SOUND L,10,8,10:GOTO 290
280 GOTO 300
290 COLOR 32:PLOT X,Y:COLOR XX+5:X=X+XX:Y=Y+YY:PLOT X,Y
300 SOUND L,L,L,L:SOUND 2,L,L,L
310 IF Z>39-RN*4 THEN GOSUB 800
315 IF STRIG(L)=L THEN GOSUB 600
320 IF RN>1 THEN GOSUB 400
390 GOTO 200
395 REM *** DIE GEISTERKETTE ***
400 XE=XE+RN/2:XZ=XZ+RN:IF XZ>20 THEN XZ=L:XP=XP+1:POKE 53248,L:POKE 53249,L:IF
XP=53250 THEN XP=53248
410 POKE XP,XE
415 IF X*8+48>XE-2 AND X*8+48<XE+10 THEN 440
420 IF XE>200 THEN XE=32
430 RETURN
440 IF XP=53248 AND Y/2=INT(Y/2) THEN 5000
450 IF XP=53249 AND Y/2<>INT(Y/2) THEN 5000
460 GOTO 420
595 REM *** SPIELER 'ZAPPT' ***
600 IF Y+KK=23 THEN RETURN
605 COLOR 12:LOCATE X,Y+KK,GG:PLOT X,Y+KK
610 FOR I=L TO 50:SOUND L,I,2,10:NEXT I:SOUND L,L,L,L:T=T-1
620 LOCATE X,Y+KK+KK,GH:IF GH=135 THEN 640
630 COLOR GG:PLOT X,Y+KK:SETCOLOR 3,7,10:RETURN
640 LJ=40:LH=2
650 LJ=LJ-8:LH=LH+3:SETCOLOR 3,7,LH:SOUND L,LJ,8,15:IF LJ=L THEN 700
660 FOR I=1 TO 50:IF STRIG(L) THEN 630
665 NEXT I:SOUND L,L,L,L
670 FOR I=1 TO 80:IF STRIG(L) THEN POP :GOTO 690
680 NEXT I:GOTO 630
690 FOR U=I TO 80:IF STRIG(L)=L THEN 650
695 NEXT U:GOTO 650
700 COLOR 170:PLOT X,Y+KK+KK:FOR I=15 TO L STEP -0.3:SOUND L,30-I,8,I:NEXT I:SET
COLOR 3,7,10
710 LL=5:COLOR 43:PLOT X,Y+KK+KK:Z=L:K=L
720 P=10:Q=10:PN=PN+50+RN*10:GOSUB 950:GOTO 630
790 REM *** ERDWESEN ***
800 MH=MH+1:IF MH=RN+1 THEN MH=L:RETURN
810 LOCATE P+PP,Q+QQ,G
815 IF G>3 AND G<7 THEN 5000
817 IF G=32 OR (G=43 AND G<>41) THEN 850
818 IF G=34 THEN ML=ML+1:IF ML=10-RN THEN ML=L:GOTO 850
820 PP1=QQ:QQ1=PP:LOCATE P+PP1,Q+QQ1,GH:IF GH=32 OR (GH=43 AND GH<>41) THEN PP=P
P1:QQ=QQ1:GOTO 850
825 LOCATE P-PP1,Q-QQ1,GH:IF GH=32 THEN PP=-PP1:QQ=-QQ1:GOTO 850
835 IF GH>3 AND GH<7 THEN 5000
840 PP=-PP:QQ=-QQ
850 COLOR 32:PLOT P,Q:P=P+PP:Q=Q+QQ:COLOR 135:PLOT P,Q
860 IF Q=Y THEN 870
```

```
865 RETURN
870 VV=(P<X)-(P>X):FOR I=1 TO 19:LOCATE P+VV*I,Q,GG
872 IF GG=34 OR GG=163 THEN COLOR 32:PLOT P+VV,Q:DRAWTO P+VV*(I-1),Q:POP :GOTO 8
85
875 IF GG>3 AND GG<7 THEN COLOR 32:PLOT P+VV,Q:DRAWTO P+VV*(I-1),Q:POP :GOTO 500
0
880 COLOR 8:PLOT P+VV*I,Q:SOUND L,255-I*10,10,10:NEXT I
885 IF I=1 THEN COLOR 34:PLOT P+VV,Q
890 RETURN
900 REM *** ERDWESEN TAUCHT AUF ***
910 FOR I=A+56 TO A+63:POKE I,L:NEXT I:SO=100:COLOR 135:PLOT P,Q
920 FOR I=7 TO L STEP -1:SO=SO-1:SOUND L,SO,10,10:RESTORE 10030:READ C
930 FOR U=I TO 7:READ C:POKE A+U+56,C:NEXT U:NEXT I:SOUND L,L,L,L
940 FOR I=45 TO L STEP -1:SOUND L,I,10,10:SETCOLOR 3,I,I:NEXT I:SETCOLOR 3,7,10:
SOUND L,L,L,L:RETURN
950 POSITION 9,2:? #6;PN:POSITION 16,2:? #6;LL:RETURN
970 ZA=ZA+1:IF ZA=25 THEN 4000
980 PN=PN+5:GOSUB 950:RETURN
995 REM *** ANFANGSGAG ***
1000 FOR I=L TO 3:POKE 53248+I,L:NEXT I:POKE 77,0
1010 GRAPHICS 20:DL=PEEK(560)+PEEK(561)*256+4:POKE DL+5,6
1020 GOTO 1070
1030 SA=PEEK(89)*256+PEEK(88):AA=SA+Y*10+X:FOR E1=1 TO LEN(T$):SETCOLOR L,E1,10:
GOSUB 1060:Z=PEEK(756)*256:Z=Z+E3*8
1040 FOR E2=L TO 7:POKE AA+E2*10,PEEK(Z+E2):SOUND D,E2*25,10,7-E2:NEXT E2:AA=AA+
11:NEXT E1:RETURN
1060 E3=ASC(T$(E1,E1))-32:RETURN
1070 X=2:Y=13:T$="CHERRY":GOSUB 1030
1080 X=3:Y=23:T$="HARRY":GOSUB 1030:FOR I=L TO 100:NEXT I
1090 X=2:Y=13:T$="       ":GOSUB 1030
1100 X=3:Y=23:T$="      ":GOSUB 1030:RETURN
1140 X=2:Y=13:T$="      ":GOSUB 1030
1150 SOUND L,U+I,10,10:NEXT U:NEXT I:SOUND L,L,L,L:RETURN
3995 REM *** RUNDE GESCHAFFT !!! ***
4000 RN=RN+1:POKE 77,0
4010 IF T<1 THEN 4040
4020 POSITION 6,6:? #6;"bonus:";T:FOR I=L TO T-1:SOUND L,40,10,10:SOUND L,L,L,L
4030 PN=PN+1:T=T-1:POSITION 9,2:? #6;PN:POSITION 12,6:? #6;T;" ";:NEXT I:T=500:F
OR I=L TO 200:NEXT I:GOTO 100
4040 FOR I=L TO 12:POSITION 6,7:? #6;"no bonus":FOR II=2 TO 10:SOUND L,II,10,10:
NEXT II:SOUND L,L,L,L
4050 POSITION 6,7:? #6;"         ":FOR II=2 TO 10:NEXT II:NEXT I
4060 POSITION 6,7:? #6;"no bonus":T=500:FOR I=1 TO 200:NEXT I:GOTO 100
4995 REM *** SPIELER KAPUTT ***
5000 FOR I=1 TO 255 STEP 15:SOUND L,I,2,10:NEXT I:COLOR 5:PLOT X,Y:POKE 77,0
5003 SETCOLOR 4,L,L:POP :SO=10:FOR I=L TO 7:RESTORE 10020:READ C:IF I THEN POKE
A+39+I,L
5005 SO=SO+20:FOR U=I TO 7:READ C:POKE A+40+U,C:NEXT U
5010 FOR U=SO TO SO+10:SOUND L,U,10,10:NEXT U:SOUND L,L,L,L
5020 FOR U=L TO 20:NEXT U:NEXT I
5040 SOUND L,L,L,L:COLOR 34:PLOT X,Y
5140 RESTORE 10015:FOR I=4 TO 6:READ H:FOR U=L TO 7:READ C:POKE A+H*8+U,C:NEXT U
:NEXT I
5160 LB=LB-1:IF LB<L THEN 5500
5170 POSITION 16,1:? #6;LB:POKE 53248,L:POKE 53249,L
5180 FOR I=255 TO 10 STEP -15:SOUND L,I,10,10:SOUND 1,I,2,10:NEXT I:SOUND L,L,L,
L:SOUND 1,L,L,L
5190 GOTO 190
5500 POSITION 5,10:? #6;" game over "
5510 FOR I=1 TO 30:GOSUB 5540:NEXT I
5520 POSITION 5,10:? #6;"           "
5530 FOR I=1 TO 30:GOSUB 5540:NEXT I:GOTO 5500
5540 IF STRIG(L) THEN RETURN
5550 POP :GOSUB 1000:GOTO 90
9995 REM *** DATA-ZEILEN ***
10000 DATA 1,255,255,255,255,255,255,255,255
10005 DATA 2,255,255,170,85,255,255,170,85
```

```
10010 DATA 3,0,6,8,8,28,62,62,28
10015 DATA 4,28,44,24,252,28,60,36,108
10020 DATA 5,28,42,149,203,126,24,60,231
10025 DATA 6,56,52,24,63,56,60,36,54
10030 DATA 7,126,219,255,171,129,129,195,126
10035 DATA 8,0,0,34,85,136,0,0,0
10050 DATA 9,255,0,255,255,0,255,255,0
10060 DATA 10,137,82,36,134,81,36,74,145
10070 DATA 11,16,24,58,110,122,62,46,8
10080 DATA 12,24,16,8,24,16,8,24,16
10190 DATA -1
10200 DATA 24,60,219,255,195,60,24,36
10300 REM *** LISTING ENDE ***
```



Autorennen für den Schneider CPC 464

Weichen Sie den gegnerischen Fahrzeugen aus!

Dieses Programm nutzt die graphischen Fähigkeiten und das schnelle Basic des CPC 464, so daß dieses Spiel trotz seines geringen Umfanges nicht so schnell langweilig wird.

Die Eingabe des Listings kann wesentlich erleichtert werden, indem die automatische Zeilennumerierung eingesetzt wird. Man muß hierbei nur „AUTO 100“ und dann die Programmzeilen ohne Zeilennummer eingeben. Sind alle Zeilen eingetippt, so muß nun die Escape-Taste gedrückt werden.

Man startet das Programm nun mit „RUN“. Sollte der Rechner während des Programmablaufs auf 80 Zeichen pro Zeile mit gelber Schrift auf blauem Grund (Programmiermodus) umschalten, so wurde bei der Eingabe ein Fehler gemacht. Man gibt nun **„PRINT ERL“** ein, um die Nummer der fehlerhaften Zeile zu erfahren.

Gespielt wird folgendermaßen:

Vor dem Start muß die Bahnbreite (1–6) gewählt werden. Ein günstiger Wert ist hier 3. Nun beginnt das Spiel sofort. Es erscheint ein Stück Straße und ein dunkelrotes Auto. Dieses Fahrzeug kann mit dem Joystick oder den Pfeiltasten gesteuert werden. Folgen Sie nun dem Verlauf der Straße und weichen Sie anderen Autos aus. Je schneller Sie fahren, desto höher ist Ihre Punktezahl. Ihr Fahrzeug läßt sich jedoch bei höherer Geschwindigkeit schlechter steuern. Das Spiel ist zu Ende, wenn Sie ein anderes Fahrzeug oder den Straßenrand rammen.

```
100 ' ########################
110 ' #                      #
120 ' #                      #
130 ' #     Autorennen       #
140 ' #     ----------       #
150 ' #                      #
160 ' #    (c) 1985  by      #
170 ' #                      #
180 ' # Juergen Baumgartl    #
190 ' #                      #
200 ' #                      #
210 ' #                      #
220 ' ########################
230 '
240 DEFINT A-Z
250 RANDOMIZE TIME
260 ON ERROR GOTO 1230
270 ON BREAK GOSUB 1230
280 MODE 1:INK 0,0:INK 1,6:INK 2,26:PAPER 0
:PEN 1:BORDER 5
290 LOCATE 15,2:PRINT "AUTORENNEN"
300 LOCATE 15,3:PRINT "----------"
310 PEN 2
320 LOCATE 15,6:PRINT CHR$(164)+" 1985  by"
330 LOCATE 12,8:PRINT "Jurgen Baumgartl"
340 LOCATE 12,10:PRINT "Am  Steinberg 13"
350 LOCATE 12,12:PRINT "6057 Dietzenbach"
360 LOCATE 12,14:PRINT "Tel: 06074/24256"
370 PEN 1
380 LOCATE 2,24:PRINT "Steuerung mit Joysti
ck oder Cursorkeys"
```

```
390 LOCATE 8,20:PRINT "Welche Bahnbreite? (1-6)"
400 WI=VAL(INKEY$):IF WI<1 OR WI>6 THEN 400
410 WI=WI+3
420 PLOT 195,287,2:PLOT 197,287:PLOT 203,287:PLOT
205,287
430 SYMBOL AFTER 254
440 SYMBOL 254,24,60,153,255,153,60,102,90
450 SYMBOL 255,126,126,24,219,255,219,24,126
460 SPEED INK 8,4
470 MODE 0
480 INK 0,13:INK 1,9:INK 2,3:BORDER 0:PAPER 1:PEN
0:CLS
490 FOR I=3 TO 15:READ J:INK I,J:NEXT I
500 DATA 0,1,2,4,5,6,7,8,9,11,16,21,24,25,26
510 SCORE!=0:X1=320:SPE=100:DIST=100:TR!=0.5:CU=0:
ST=320-WI*16:LM=20:RM=620-WI*32:ST$=STRING$(WI,CHR
$(143)):WI=(WI-1)*32
520 PLOT -10,-10,0
530 TAG:FOR I=0 TO 400 STEP 16:MOVE ST,I:PRINT ST$
;:NEXT I:TAGOFF
540 EVERY 200 GOSUB 1010
550 ' MAIN LOOP
560 IF (NOT (INKEY(0) AND INKEY(72))) AND SPE>=5 T
HEN SPE=SPE-5
570 IF (NOT (INKEY(2) AND INKEY(73))) AND SPE<=95
THEN SPE=SPE+5
580 IF NOT (INKEY(8) AND INKEY(74)) THEN GOSUB 980
:GOSUB 920:X1=X1-4-SPE\5:GOSUB 980:GOSUB 860
590 IF NOT (INKEY(1) AND INKEY(75)) THEN GOSUB 980
:GOSUB 920:X1=X1+4+SPE\5:GOSUB 980:GOSUB 860
600 GOSUB 680
610 T1=TEST(X1,51):T2=TEST(X1+9,51):T3=TEST(X1+16,
51):T4=TEST(X1+28,51)
620 IF T1<>0 OR T2<>0 OR T3<>0 OR T4<>0 THEN 1050
630 SOUND 132,SPE*2+400,1000,12
640 SOUND 129,400,1000,4,,1,30
650 FOR I=0 TO SPE:NEXT I
660 SCORE!=SCORE!+110-SPE
670 GOTO 550
680 ' SCROLL DOWN
690 GOSUB 980
700 GOSUB 920
710 LOCATE 1,1:PRINT CHR$(8)
720 GOSUB 860
730 '
740 A!=RND
750 IF A!<TR! AND ST>LM THEN ST=ST-CU
760 IF A!>TR! AND ST<RM THEN ST=ST+CU
770 TAG
780 PLOT -10,-10,0:MOVE ST,399:PRINT ST$;
790 TAGOFF
800 DIST=DIST+1
810 IF RND>0.2 OR DIST<5 THEN RETURN
820 X=ST+INT(RND*WI+0.5):Y=399:C=INT(RND*12)+3
```

```
830 GOSUB 860
840 DIST=0
850 RETURN
860 ' PRINT CAR
870 TAG
880 PLOT -10,-10,C:MOVE X,Y:PRINT CHR$(254);
890 MOVE X,Y-16:PRINT CHR$(255);
900 TAGOFF
910 RETURN
920 ' REMOVE CAR
930 TAG
940 MOVE X,Y:PEN 0:PRINT " ";
950 MOVE X,Y-16:PRINT " ";
960 TAGOFF
970 RETURN
980 ' PLAYER'S CAR
990 C=2:X=X1:Y=47
1000 RETURN
1010 '
1020 TR!=RND
1030 IF ABS(TR!-0.5)<0.2 THEN CU=4 ELSE CU=8
1040 RETURN
1050 ' CRASH
1060 BORDER 26,6
1070 INK 0,13,0
1080 INK 1,22,9
1090 FOR I=1 TO 31
1100 SOUND 135,0,500,15-I\10,0,0,I
1110 FOR J=1 TO I*10:NEXT J
1120 NEXT I
1130 FOR I=1 TO 5000:NEXT I
1140 PAPER 2:PEN 5:CLS:BORDER 0
1150 LOCATE 5,5:PRINT "Ihre Wertung:"
1160 LOCATE 4,7:PRINT USING "########";SCORE!;:PRIN
T " PUNKTE"
1170 PEN 4
1180 LOCATE 4,20:PRINT "Zum  Fortfahren"
1190 LOCATE 4,22:PRINT "Taste druecken!"
1200 WHILE INKEY$<>"":WEND
1210 WHILE INKEY$="":WEND
1220 RUN
1230 ' ERROR/BREAK
1240 INK 0,1:INK 1,24:BORDER 1:PAPER 0:PEN 1:MODE
2
1250 END
```

Zahlenwerte des Zeichens, Änderungen und Fehlerkorrekturen können jederzeit vorgenommen werden.

Tastaturbelegung

Jeder Programmierer hat sich schon mal eine andere Belegung der Tastatur gewünscht. Kein Problem für dieses Programm. Basic-Token, Blockgraphik, griechische Buchstaben oder Sonderzeichen lassen sich wahlweise auf die „normal", „shift" oder „ctrl"-Ebene der Tastatur legen. Mit diesem Programm läßt sich Programmieren komfortabel gestalten.

Spritegenerator

Bewegte Objekte gehören zu einem Spiel wie der Bildschirm zum Computer. Diese Sprites lassen sich einfach und bequem in verschiedenen Größen und Farben auf dem Bildschirm entwerfen. Jeder einzelne Punkt kann frei definiert werden, das Zeichen wird zur Kontrolle am Bildschirmrand dargestellt. Per Tastendruck wird der Sprite abgespeichert und kann in eigene Programme eingebunden werden.

„Graphic Utilities 464

Dieses Programmpaket ist eine praktische Hilfe für die Programmierung in Basic auf dem **Schneider CPC 464.** Zeichenerzeugung, Tastaturbelegung mit Basic-Tokens oder Graphiksymbolen und ein Spritegenerator erleichtern dem Anfänger als auch dem Profi die Programmierung eigener Programme und Spiele.

Zeichenerzeugung

Wer schon einmal einen eigenen Zeichensatz erstellt hat, weiß wie mühselig diese Arbeit mit Bleistift, Papier und Taschenrechner ist. Anders mit dem Zeichenerzeugungsprogramm. In einer Matrix werden mit den Cursor Punkte gesetzt, auf Tastendruck erscheinen die

Dynamics bietet das Programmpaket „Graphic Utilities 464" im Rahmen einer neuen, preisgünstigen Softwarereihe für den Schneider CPC 464 an. Diese Reihe besteht aus verschiedenen Archivprogrammen, einem Graphikprogramm „Diagramm 464" und einem Vokabeltrainer.

Empf. Endverbraucherpreis auf Kassette 49,– DM.

Maschinencoderoutinen für ZX Spectrum 48K

Diese Routinen ermöglichen das Vergrößern von Buchstaben, das Scrollen nach oben als auch nach unten

Die folgenden Routinen eignen sich zum Einbau in eigene Programme (Anleitungen, Scoreanzeigen etc.). Sie funktionieren folgendermaßen:

Routine zur Vergrößerung von Buchstaben
Sie beginnt bei Adresse 64000 und reicht bis 64151. Bevor Sie die Routine starten, müssen Sie die ASCII-Codes der Buchstaben (Handbuch Anhang A) in die Adressen 64152ff. poken. Der letzte Code muß eine Null sein; daran erkennt der Computer, daß der Text zu Ende ist. Ferner muß in Adresse 64008 die y-Koordinate und in Adresse 64012 die x-Koordinate der ersten Print-Position stehen. Durch „RANDOMIZE USR 64000" startet man die Routine. Noch ein Vorteil des Programms: Falls der Text über den Bildschirm hinausgeht, bricht die Routine ab, um einen Absturz zu vermeiden. Die Routine verwendet den Zeichensatz, auf den die Systemvariable CHARS zeigt. Das heißt: Auch eigens definierte Zeichensätze können mit dieser Routine verwendet werden, wenn man in CHARS die Adresse des neuen Zeichensatzes pokt.

Scroll up
Diese Routine belegt die Adressen 63800 bis 63890. Folgende Werte müssen vor dem Start gepokt werden:
In 63801 steht die x-, in 63802 die y-Koordinate der oberen linken Print-Position. In 63804 steht die Anzahl der Spalten (nach rechts gezählt), in 63805 die Anzahl der Zeilen (nach unten). In der Adresse 63823 steht eine Zahl, die angibt, wie viele Pixels nach oben gescrollt wird (bei einem einzigen Anruf des Maschinenprogramms durch „RANDOMIZE USR 63800").

Scroll down
Diese Routine ist analog zur vorherigen. Sie besetzt die Speicherplätze von 63900 bis 63993. Pokes:
63901 x-Koord. der Print-Position oben links
63902 y-Koord. der Print-Position oben links
63904 Anzahl der Spalten
63905 Anzahl der Zeilen
63926 (!) Anzahl, um wieviel Pixels nach untern gescrollt wird.
Start durch „RANDOMIZE USR 63900".
Die beiden Scrollroutinen lassen sich leicht auf niedrigere Adressen verschieben (für 16K), da nur relative Sprünge vorkommen. Mit diesen beiden Routinen können einzelne Print-Positionen, Zeilenabschnitte, Spaltenabschnitte oder ganze Blöcke, natürlich auch der ganze Bildschirm gescrollt werden.

Tippen Sie zuerst das Listing 2 ein, mit dessen Hilfe Sie dann die Maschinencoderoutinen eingeben.
Anschließend geben Sie NEW ein, wobei das Maschinenprogramm nicht gelöscht wird, und geben nun das Demoprogramm ein. Mit „Goto 9990" werden die Programme auf Kassette abgesavet.

Scroll up

```
63800  33   0   0   1   1
63805   1 125 230 248 198
63810  64  87 125 230   7
63815  15  15  15 132  95
63820 111  98  62   1 245
63825 197 229 213 197   6
63830   7 197  36 229   6
63835   0 237 176 225  84
63840  93 193  16 243 193
63845  16  14  65  54   0
63850  35  16 251 209 225
63855 193 241  61  32 220
63860 201 125 254 224  48
63865   9 198  32 111 124
63870 214   7 103  24   4
63875  36 214 224 111 197
63880 229   6   0 237 176
63885 225  84  93 193  24
63890 193   0   0   0   0
```

Scroll down

```
63900  33   0   0   1   1
63905   1 125 128  61 111
63910 230 248 198  71  87
63915 125 230   7  15  15
63920  15 132  95 111  98
63925  62   1 245 197 229
63930 213 197   6   7 197
63935  37 229   6   0 237
63940 176 225  84  93 193
63945  16 243 193  16  14
63950  65  54   0  35  16
63955 251 209 225 193 241
63960  61  32 220 201 125
63965 254  32  56   9 214
63970  32 111 124 198   7
63975 103  24   4  37 198
63980 224 111 197 229   6
63985   0 237 176 225  84
63990  93 193  24 193   0
```

Vergrößerung von Buchstaben

```
64000 221  33 152 250  33
64005   0  64  17   0   0
64010  25  30   0 123   6
64015   2 222   8  56   9
64020 245 124 198   8 103
64025 241  95  16 243  62
64030   0 187  40   7   1
64035  32   0   9  29  24
64040 246 237  75  54  92
64045 221  94   0  62   0
64050 187 200 229  38   0
64055 107  41  41  41   9
64060 235 225 229 229 205
64065 133 250 225 125   1
64070 224  32 185  40   9
64075  12  16 250   1  32
64080   0   9  24   7 214
64085 224 111 124 198   8
```

```
64090  103 205 133 250 225
64095  125 230  31 254  31
64100   32  26 125 230 223
64105  254 223  32  15 230
64110   80 254  80 200 124
64115  198   8 103 125 214
64120  223 111  24   5   1
64125   32   0   9  35 221
64130   35  24 164   6   8
64135   62   1 245  26 119
64140   36 241 238   1 254
64145    0  40   1  19  16
64150  242 201   0   0   0
```

Listing 2

zur Eingabe des Maschinenprogramms!

```
  5 CLEAR 63799
 10 FOR f=1 TO 3
 20 READ a,b,sum:LET pr=0
 30 FOR g=a TO b
 40 INPUT (g);": ";p
 50 PRINT g,p: POKE g,p
 60 LET pr=pr+p
 70 NEXT g
 80 IF sum=pr THEN NEXT f: STOP
 90 PRINT "Fehler im Maschinenp
rogramm":STOP
100 DATA 63800,63890,10532
105 DATA 63900,63993,10641
110 DATA 64000,64151,15496
```

Listing des Demoprogrammes

```
  10 CLEAR 63799: LOAD ""CODE
  20 BORDER 0: PAPER 0: BRIGHT 1
: CLS : POKE 23624,69: INPUT " "
  30 DIM k$(64)
  40 FOR f=0 TO 20 STEP 2: PRINT
 AT f,0; INK (f-INT (f/14)*14)/2
+1;k$: NEXT f
  50 LET x=64012: LET y=64008: L
ET a=64152
  60 LET p1=63823: LET p2=63926
  70 LET x1=63801: LET y1=63802:
 LET s1=63804: LET z1=63805
  80 LET x2=x1+100: LET y2=y1+10
0: LET s2=s1+100: LET z2=z1+100
  90 POKE p1,8: POKE p2,8
 100 GO SUB 520
 110 GO SUB 470
 120 POKE x1,0: POKE y1,5: POKE
z1,24
 130 GO SUB 210
 140 GO SUB 230
 150 GO SUB 470
 160 LET i=2: POKE y,5: POKE x,1
1: FOR f=64174 TO 64184: POKE f,
0: NEXT f: GO SUB 310
 170 POKE x1,13: POKE y1,5: POKE
 s1,22: POKE z1,11: POKE p1,6
 180 GO SUB 350: POKE x1,0: POKE
 z1,24: RANDOMIZE USR 63800
 190 GO SUB 380
 200 PAUSE 0: GO TO 9999
 210 FOR f=1 TO 22: POKE s1,f: R
ANDOMIZE USR 63800: NEXT f
 220 RETURN
 230 POKE x2,0: POKE y2,5: POKE
s2,11: POKE z2,13
 240 POKE x1,11: POKE y1,16: POK
E z1,12
 250 RANDOMIZE USR 63900
 260 FOR f=10 TO 1 STEP -1
 270 POKE s2,f: POKE y1,PEEK y1+
1: POKE s1,f
 280 RANDOMIZE USR 63800: RANDOM
IZE USR 63900
 290 NEXT f
 300 RETURN
 310 FOR g=1 TO 5: RESTORE 560
 320 FOR f=64163 TO 64163+i: POK
E f,32: NEXT f: FOR f=64163+i TO
 64173+i: READ z: POKE f,z: NEXT
 f
 330 LET i=i+2: POKE y,PEEK y-1:
 RANDOMIZE USR 64000: NEXT g
 340 RETURN
 350 FOR g=1 TO 11: FOR f=64153
TO 64163: POKE f-1,PEEK f: NEXT
f: FOR f=64182 TO 64172 STEP -1:
 POKE f+1,PEEK f: NEXT f
 360 RANDOMIZE USR 64000: RANDOM
IZE USR 63800: NEXT g
 370 RETURN
 380 POKE p1,104: RANDOMIZE USR
63800
 390 GO SUB 520
 400 FOR g=160 TO 16 STEP -16: P
OKE p1,g: POKE x1,PEEK x1+2: POK
E z1,PEEK z1-2
 410 GO SUB 470: RANDOMIZE USR 6
3800: NEXT g
 420 GO SUB 470: POKE y,5: POKE
64174,0
 430 FOR f=21 TO 0 STEP -1: POKE
 x,f: RANDOMIZE USR 64000: NEXT
f
 440 FOR f=1 TO 22: POKE x,f: RA
NDOMIZE USR 64000: PRINT AT f-1,
5;"                    ": NEXT
 f
 450 POKE x2,22: POKE y2,5: POKE
 s2,22: POKE z2,2: POKE p2,16: R
ANDOMIZE USR 63900
```

```
 460 RETURN
 470 POKE x,22: POKE y,31
 480 FOR f=1 TO 27
 490 RANDOMIZE USR 64000: POKE y
,PEEK y-1
 500 NEXT f
 510 RETURN
 520 RESTORE 550
 530 FOR f=1 TO 27: READ z: POKE
 a+f-1,z: NEXT f
 540 RETURN
 550 DATA 68,69,77,79,78,83,84,8
2,65,84,73
 560 DATA 79,78,83,80,82,79,71,8
2,65,77,77,32,32,32,32,32
9990 SAVE "demo" LINE 10: SAVE "
demo1"CODE 63800,355
9995 VERIFY "": VERIFY ""CODE
```

Assemblerlistings

```
 10 ;ROUTINE FUER GROSSE
    ;BUCHSTABEN
 20         ORG 64000
 30 CHARS   EQU 23606
 40         LD  IX,ADDR
 50         LD  HL,16384
 60         LD  DE,0
 70         ADD HL,DE
 80         LD  E,0
 90         LD  A,E
100         LD  B,2
110 BLOCK   SBC A,8
120         JR  C,BL1
130         PUSH AF
140         LD  A,H
150         ADD A,8
160         LD  H,A
170         POP AF
180         LD  E,A
190         DJNZ BLOCK
200 BL1     LD  A,0
210 ZEILE   CP  E
220         JR  Z,ZEIL1
230         LD  BC,32
240         ADD HL,BC
250         DEC E
260         JR  ZEILE
270 ZEIL1   LD  BC,(CHARS)
280         LD  E,(IX)
290         LD  A,0
300         CP  E
310         RET Z
320         PUSH HL
330         LD  H,0
340         LD  L,E
350         ADD HL,HL
360         ADD HL,HL
370         ADD HL,HL
380         ADD HL,BC
390         EX  DE,HL
400         POP HL
410         PUSH HL
420         PUSH HL
430         CALL ZEICHN
440         POP HL
450         LD  A,L
460         LD  BC,8416
470 NXTBL   CP  C
480         JR  Z,NXBL1
490         INC C
500         DJNZ NXTBL
510         LD  BC,32
520         ADD HL,BC
530         JR  NXLINE
540 NXBL1   SUB 224
550         LD  L,A
560         LD  A,H
570         ADD A,8
580         LD  H,A
590 NXLINE  CALL ZEICHN
600         POP HL
610         LD  A,L
620         AND 31
630         CP  31
640         JR  NZ,SPALTE
650         LD  A,L
660         AND 223
670         CP  223
680         JR  NZ,NXLIN2
690         AND 80
700         CP  80
710         RET Z
720         LD  A,H
730         ADD A,8
740         LD  H,A
750         LD  A,L
760         SUB 223
770         LD  L,A
780         JR  ENDE
790 NXLIN2  LD  BC,32
800         ADD HL,BC
810 SPALTE  INC HL
820 ENDE    INC IX
830         JR  ZEIL1
840 ZEICHN  LD  B,8
850         LD  A,1
860 NXCHAR  PUSH AF
870         LD  A,(DE)
880         LD  (HL),A
890         INC H
900         POP AF
910         XOR 1
920         CP  0
930         JR  Z,CHREND
940         INC DE
950 CHREND  DJNZ NXCHAR
960         RET
970 ADDR    END
```

Assemblerlisting Scroll:

```
 10 ;SCROLL UP
 20 ;HL:=X/Y KOORD.DER OBERST
    ;EN LINKSTEN ZELLE
 30 ;B=ZEILENANZAHL
 40 ;C=SPALTENANZAHL
 50 ;
 60         ORG  63800
 70         LD   HL,0
 80         LD   BC,257
 90         LD   A,L
100         AND  248
110         ADD  A,64
120         LD   D,A
130         LD   A,L
140         AND  7
150         RRCA
160         RRCA
170         RRCA
180         ADD  A,H
190         LD   E,A
200         LD   L,A
210         LD   H,D
220 ;A SPEICHERT ANZAHL DER
230 ;PIXEL,DIE IN EINEM LAUF
240 ;GESROLLT WIRD
250 ;
```

```
260         LD   A,1
270 PIXLU   PUSH AF
280         PUSH BC
290         PUSH HL
300         PUSH DE
310 ZEILU   PUSH BC
320         LD   B,7
330 REIHU   PUSH BC
340         INC  H
350         PUSH HL
360         LD   B,0
370         LDIR
380         POP  HL
390         LD   D,H
400         LD   E,L
410         POP  BC
420         DJNZ REIHU
430         POP  BC
440         DJNZ NEXTU
450         LD   B,C
460 NXTU2   LD   (HL),0
470         INC  HL
480         DJNZ NXTU2
490         POP  DE
500         POP  HL
510         POP  BC
520         POP  AF
530         DEC  A
540         JR   NZ,PIXLU
550         RET
560 NEXTU   LD   A,L
570         CP   224
580         JR   NC,BLOKU
590         ADD  A,32
600         LD   L,A
610         LD   A,H
620         SUB  7
630         LD   H,A
640         JR   LDIRU
650 BLOKU   INC  H
660         SUB  224
670         LD   L,A
680 LDIRU   PUSH BC
690         PUSH HL
700         LD   B,0
710         LDIR
720         POP  HL
730         LD   D,H
740         LD   E,L
750         POP  BC
760         JR   ZEILU
770         DEFS 9
780 ;
790 ;SCROLL DOWN
800 ;ANALOG SCROLL UP
810 ;
820         LD   HL,0
830         LD   BC,257
840         LD   A,L
850         ADD  A,B
860         DEC  A
870         LD   L,A
880         AND  248
890         ADD  A,71
900         LD   D,A
910         LD   A,L
920         AND  7
930         RRCA
940         RRCA
950         RRCA
960         ADD  A,H
970         LD   E,A
980         LD   L,A
990         LD   H,D
1000        LD   A,1
1010 PIXLD  PUSH AF
1020        PUSH BC
1030        PUSH HL
1040        PUSH DE
1050 ZEILD  PUSH BC
1060        LD   B,7
1070 REIHD  PUSH BC
1080        DEC  H
1090        PUSH HL
1100        LD   B,0
1110        LDIR
1120        POP  HL
1130        LD   D,H
1140        LD   E,L
1150        POP  BC
1160        DJNZ REIHD
1170        POP  BC
1180        DJNZ NEXTD
1190        LD   B,C
1200 NXTD2  LD   (HL),0
1210        INC  HL
1220        DJNZ NXTD2
1230        POP  DE
1240        POP  HL
1250        POP  BC
1260        POP  AF
1270        DEC  A
1280        JR   NZ,PIXLD
1290        RET
1300 NEXTD  LD   A,L
1310        CP   32
1320        JR   C,BLOKD
1330        SUB  32
1340        LD   L,A
1350        LD   A,H
1360        ADD  A,7
1370        LD   H,A
1380        JR   LDIRD
1390 BLOKD  DEC  H
1400        ADD  A,224
1410        LD   L,A
1420 LDIRD  PUSH BC
1430        PUSH HL
1440        LD   B,0
1450        LDIR
1460        POP  HL
1470        LD   D,H
1480        LD   E,L
1490        POP  BC
1500        JR   ZEILD
1510        END
```

Variablenliste des DEMO:

a=64152 Addresse der Codes für grosse Buchstaben
f,g Schleifenzähler
i Zähler
p1=63823 Addresse für scroll up
p2=63926 Addresse für scroll down
s1=63804 Spaltenanzahl
s2=63904 Spaltenanzahl
x1=63801 x-Koord. scroll up
x2=63901 x-Koord. scroll down
y1=63802 y-Koord. scroll up
y2=63902 y-Koord. scroll down
z1=63805 Anzahl der Zeilen für scroll up
z2=63905 Anzahl der Zeilen für scroll down
k$ Leerstring

Database für ZX-Spectrum 48 K

Ein flexibles Dateiprogramm zur Erstellung Ihres eigenen Datenprogrammes

Das Programm **„DATABASE"** ist in BASIC geschrieben, was in diesem Falle eine genügende Ausführungsgeschwindigkeit bietet.

Dieses Programm definiert seine eigenen Befehle und Meldungen, welche den Benutzer befähigen, eine Datenbasis in automatisch alphabetischer Reihenfolge und fortlaufender Numerierung zu erstellen. Das Programm offeriert die Möglichkeiten Eintragungen vorzunehmen, zu korrigieren, aufzulisten und auszudrucken, zu löschen, und was den Sinn ausmacht, jedes beliebige Textfragment in beliebig vorwählbaren Abschnitten der Datei zu suchen, wobei ein Limit und der Ausgabemodus prädefiniert werden können.
Die wichtigsten gerade ausführbaren Kommandos und Spezifikationsmöglichkeiten werden in einem Menü dargestellt, so daß der Benutzer, ohne sich Kommandos merken zu müssen, mit dem Programm zurechtkommt.

„DATABASE" ist zur Verarbeitung einer Literaturdatei, o. ä. gedacht. Jeder Eintrag ist in vier Untereinträge „AUTHOR(S)", „TITLE", „SOURCE", „SIGNATURE/DIVERSE" unterteilt. Im **EDITOR Modus** kann man mittels eines beweglichen Fensters zwischen diesen Untereinheiten wählen, welche dann im INVERSE VIDEO aufleuchten. Dies ist immer bei einer leeren Eingabe der Fall, ein „X" oder „x" führt zum Verlassen des EDITORS.

In dieser Form sind 118 Dateneintragungen mit einer Länge von 260 Zeichen möglich. Eintragungen benötigen den Bruchteil einer Sekunde, Suchdurchläufe bis zu vier Minuten für die ganze Datei, plus ggf. Druckerzeit bei mehreren Treffern.

Das Programm ist so aufgebaut, daß die meist durchlaufenen Unterprogramme über den anderen Programmteilen im RAM stehen. Jedes spezifische Programmstück ist mit einer erklärenden REMARK versehen. Um die Lesbarkeit der ansonsten konfusen Sinclair-Listings zu verbessern, habe ich durch Einfügen von Blanks die Programmzeilen isoliert und verschiedene Prozeduren eingerückt.
Somit ist der Benutzer leicht in der Lage, das Programm zu lesen und auch Veränderungen im Layout des Eintragungsschemas nach seinen eigenen Bedürfnissen vorzunehmen, insbesondere anderen Dateiformen anzupassen.
Sollte das Programm unterbrochen werden, so hat eine Fortsetzung mit **CONTINUE** oder **GOTO 5000** im Hauptprogramm zu erfolgen. **RUN** bewirkt einen **RESTART** und **löscht die alte Datei.** Normalerweise wird man bei der Benutzung nicht aus dem Programm aussteigen, da alle Befehle von demselben gestellt werden, auch SAVE, VERIFY, LOAD, RESTART, etc. mit deren Spezifikationen.

Kommandotafel und Spezifikationen

EXIT
zurück zum Hauptmenü; wie GOTO 5000

PRINT
Ausgabe auf dem Bildschirm und/oder Drucker
(START) – A oder a oder Zahl für Anfang
(END) – E oder e oder Zahl für Ende
(PRINT Y/N?) – Y oder y für zusätzlichen Druckeroutput

SEARCH
SUCHROUTINE, Wahlmöglichkeit innerhalb von AUTHOR(S), TITLE, SOURCE, SIGNATURE
(ITEM TO SEARCH FOR) – beliebiges Textfragment
(START) – A oder a oder Zahl für Anfang
(END) – E oder e oder Zahl für Ende
(MAX. HITS?) – Limit für die Suche nach einem Textstück
(PRINTER OUTPUT?) – Y oder y für zusätzlichen Druckeroutput, bei automatischer Fortsetzung der Suche

ENTER
Einstieg in den EDITOR
ENTER-TASTE – Verschieben des Fensters, wenn keine Eingabe
– Verlassen des EDITORS, Eingabe = „EXIT"
– Eintragung in die Datei, zurück zum Hauptmenü, Eingabe = „X", „x"
– Eintragung in das Fenster des EDITORS bei irgendwelchen anderen Eingaben.

ERASE
Löschen und Neuordnen der Datei
(NO.), NO oder NO., um per Eintragungsnummer zu löschen
(SIGNATURE), Textstück zu suchen und ggf. zu löschen
„DATABASE" zeigt zunächst die spezifizierte Eintragung und fragt nach Bestätigung mit (CONFIRM ERASE). Nur Y oder y zum Löschen.

CORREKT
Editieren einer Eintragung über ihre Ordnungsnummer, ggf. zuerst Suchen der zu korrigierenden Textstelle mit SEARCH.
(NO. TO CORRECT) – Eingabe der Ordnungsnummer

PAUSE-Taste
Unterbrechung der Druckerausgabe, ANY KEY TO CONTINUE

BREAK-Taste
Wirkung wie EXIT, z. B. während der Ausgabe.

SAVE
SAVE die jetzige Datei auf Magnetband
(CHANGE FILE NAME?) – Y oder y um den Datennamen zu verändern
(ENTER NAME FOR DATA) – Bis zu 10 Stellen langen Datennamen eingeben.

VERIFY
VERIFY ein Magnetband gegen die momentane Datei; automatisch nach SAVE!

WICHTIG

Alle Kommandos müssen in **GROSS-BUCHSTABEN** geschrieben werden **(CAPSLOCK).** Alle Spezifikationen können auch in Kleinbuchstaben erfolgen. Alle Kommandos dürfen bis auf drei Buchstaben abgekürzt werden. Letzteres kann der Benutzer auch in Programmzeile 5000 ff variieren! Nach einer Programmunterbrechung in das Sinclair Betriebssystem bitte mit CONTINUE oder GOTO 5000 das Hauptmenü aufrufen!

MELDUNGEN:

RESTART OR LOADING DATA?
SPECIFY:
ENTER FILE NAME:
ENTER NAME OF FILE TO LOAD:
CHANGE FILE NAME?
ENTER NAME FOR DATA:
CATION
NO SUCH COMMAND
SYNTAX ERROR
WORKING
DATABASE EMPTY
PRINT SEARCH ENTER ERASE CORRECT
START? END? PRINTER Y/N?
ENTER ITEM TO SEARCH FOR:
FAULTY PARAMETER SPECIFI-
START? END? MAX. HITS?
PRINTER OUTPUT?
READY SEARCHING WITH (NO) HIT(S)
HIT ANY KEY TO CONTINUE SEARCHING
INTERRUPT
ERASE PER NO./SIGNATURE?
CONFIRM ERASE
ENTER NO. TO ERASE:
ENTER SIGNATURE:
NEGATIVE ERASE – NO SUCH ITEM WITH SIGNATURE (SIGNATURE)
NO. TO CORRECT?
REWIND TAPE FOR VERIFYING
POSITIVE VERIFYING

```
      STEFANEC COMPUTING


           PRESENTS


        D A T A B A S E


   1 REM D A T A B A S E
 100 DIM A$(118,80)
 101 DIM B$(118,80)
 102 DIM C$(118,50)
 103 DIM D$(118,50)
 104 DIM E(1)
 110 CLS
 120 PAPER 1: BORDER 1: BRIGHT
     1: INK 7: CLS
 130 PRINT FLASH 1; INK 6;AT 10,
     7;"STEFANEC COMPUTING":
     PAUSE 50
 140 PRINT FLASH 1; INK 6;AT 14,
     11;"PRESENTS": PAUSE 50
 150 PRINT FLASH 1; INK 6;AT 18,
     8;"D A T A B A S E"
 160 PRINT FLASH 1; INK 6;AT 10,
     7;"STEFANEC COMPUTING";AT
     14,11;"PRESENTS";AT 18,8;
     "D A T A B A S E"
 190 PAUSE 500
 199
 200 REM  LOAD FILE/RESTART
 201
 202 INPUT "RESTART OR LOADING";
     " DATA ?"'"SPECIFY >"; LINE
     Z$
 204 IF LEN Z$<3 THEN GO SUB
     3200: GO TO 202
 206 IF Z$(1 TO 3)="RES" THEN
     LET EOF=1: INPUT "ENTER FI"
     ;"LE NAME >"; LINE N$: IF
     LEN N$>10 THEN GO SUB 3200
     : GO TO 206
 208 IF Z$(1 TO 3)="RES" THEN
     CLS : GO TO 5000
 209 IF Z$(1 TO 3)="LOA" THEN
     GO TO 215
 210 GO SUB 3100: GO TO 202
 215 INPUT "ENTER NAME OF FILE "
     ;"TO LOAD >"'TAB 20; LINE
     N$
 220 LOAD N$ DATA A$()
 230 LOAD N$ DATA B$()
 240 LOAD N$ DATA C$()
 250 LOAD N$ DATA D$()
 260 LOAD N$ DATA E()
 270 LET EOF=E(1)
 280 CLS
 290 GO SUB 4000
 300 GO TO 5000
 400
1000 REM COMPARISON
1001
1010 LET T=LEN T$
1020 LET X=LEN X$
1030 LET L=X-1
1040 FOR I=1 TO T-L
1050   IF X$=T$(I TO I+L) THEN
       GO SUB 5580
1060 NEXT I
1070 RETURN
1071
2000
2010 FOR K=A TO E
2020   LET T$=A$(K): GO SUB 1000
2025 CLS : PRINT "WORKING":
     PAUSE 20: CLS
2026 IF INKEY$=" " THEN GO TO
     5000
2030 NEXT K
2040 RETURN
2100
2110 FOR K=A TO E
2120   LET T$=B$(K): GO SUB 1000
2125 CLS : PRINT "WORKING":
     PAUSE 20: CLS
2126 IF INKEY$=" " THEN GO TO
     5000
2130 NEXT K
2140 RETURN
2200
2210 FOR K=A TO E
2220   LET T$=C$(K): GO SUB 1000
2225 CLS : PRINT "WORKING":
     PAUSE 20: CLS
2226 IF INKEY$=" " THEN GO TO
     5000
2230 NEXT K
2240 RETURN
2300
2310 FOR K=A TO E
2320   LET T$=D$(K): GO SUB 1000
2325 CLS : PRINT "WORKING":
     PAUSE 20: CLS
2326 IF INKEY$=" " THEN GO TO
     5000
2330 NEXT K
2340 RETURN
2400
3000 REM  STATEMENTS
3001
3010 PRINT #1; FLASH 1;"FAULTY";
     " PARAMETER SPECIFICATION":
     PAUSE 100
3020 RETURN
3100 PRINT #1; FLASH 1;
     "NO SUCH COMMAND": PAUSE 99
3110 RETURN
3200 PRINT #1; FLASH 1;"SYNTAX "
     ;"ERROR": PAUSE 99
3210 RETURN
3220
3700 REM EDITOR
3701
3714 LET K=0
3720 LET FGA=1: GO SUB 4100
3722 INPUT LINE Z$
3723 IF Z$="EXIT" THEN GO TO 5E3
```

```
3725 IF Z$="X" OR Z$="x" THEN
     RETURN
3726 IF Z$<>"" THEN
       LET A$(1)=Z$: GO SUB 4100
3728
3730 LET FGB=1: GO SUB 4100
3732 INPUT LINE Z$
3733 IF Z$="EXIT" THEN GO TO 5E3
3735 IF Z$="X" OR Z$="x" THEN
     RETURN
3736 IF Z$<>"" THEN
       LET B$(1)=Z$: GO SUB 4100
3738
3740 LET FGC=1: GO SUB 4100
3742 INPUT LINE Z$
3743 IF Z$="EXIT" THEN GO TO 5E3
3745 IF Z$="X" OR Z$="x" THEN
     RETURN
3746 IF Z$<>"" THEN
       LET C$(1)=Z$: GO SUB 4100
3748
3750 LET FGD=1: GO SUB 4100
3752 INPUT LINE Z$
3753 IF Z$="EXIT" THEN GO TO 5E3
3755 IF Z$="X" OR Z$="x" THEN
     RETURN
3756 IF Z$<>"" THEN
       LET D$(1)=Z$: GO SUB 4100
3758
3760 GO TO 3720
3770
4000 REM DISPLAY&PROMPT
4001
4010 LET A$(1)="..................
.....................................
:....................................
"
4020 LET B$(1)="..................
.....................................
:....................................
"
4030 LET C$(1)="..................
...:.................................
:.."
4040 LET D$(1)="..................
...:.................................
:.."
4050 LET FGA=0: LET FGB=0:
     LET FGC=0: LET FGD=0:
     LET FGP=0
4060 RETURN
4061
4100 CLS
4101 LET K=K+1
4102 IF K<>1 THEN PRINT AT 3,0;
     INVERSE (K=1);"NO.= ";K-1;
     TAB 16;"FILE: ";N$: GO TO
     4110
4104 PRINT AT 3,0; INVERSE 1;"ED
ITOR MODE"
4110 PRINT AT 5,0;"AUTHOR(S):"'
     INVERSE (FGA=1);A$(K)'':
     LET FGA=0
4120 PRINT "TITLE:"' INVERSE (FG
     B=1);B$(K)'': LET FGB=0
4130 PRINT "SOURCE:"' INVERSE (F
     GC=1);C$(K)'': LET FGC=0
4140 PRINT "SIGNATURE/DIVERSE:"'
     INVERSE (FGD=1);D$(K)': LET
     FGD=0
4150 IF FGP=1 THEN COPY
4155 LET K=K-1
4160 RETURN
4161
5000 REM INPUT&PROCEDURES
5001
5005 IF EOF=1 THEN PRINT #1;
     FLASH 1;"DATABASE EMPTY ":
     PAUSE 100
5010 LET Z$="    "
5020 INPUT
     "PRINT SEARCH ENTER ERASE "
     ;"CORRECT"'"GIVE COMMAND >"
     ; LINE Z$
5030 IF LEN Z$<3 THEN GO TO 5200
5040 IF Z$(1 TO 3)="PRINT"(1 TO
      3) THEN GO TO 5500
5050 IF Z$(1 TO 3)="SEARCH"(1 TO
     3) THEN GO TO 5600
5060 IF Z$(1 TO 3)="ENTER"(1 TO
      3) THEN GO TO 5700
5070 IF Z$(1 TO 3)="ERASE"(1 TO
      3) THEN GO TO 5800
5080 IF Z$(1 TO 3)="CORRECT"(1
     TO 3) THEN GO TO 5900
5100 IF Z$(1 TO 3)="SAVE"(1 TO
     3) THEN GO TO 9000
5110 IF Z$(1 TO 3)="VERIFY"(1 TO
     3) THEN GO TO 9100
5200 GO SUB 3100: GO TO 5000
5201
5500 REM PRINT
5501
5505 IF EOF=1 THEN GO TO 5005
5510 INPUT TAB 9;
     "START,END,PRINT Y/N?"'
     "SPECIFY >"; LINE G$;TAB 15
     ; LINE H$;TAB 19; LINE F$
5520 IF F$="Y" OR F$="y" THEN
     LET FGP=1
5525 LET FG1=0: LET FG2=0
5530 IF G$="A" OR G$="a" THEN
     LET A=1: LET FG1=1
5534 IF H$="E" OR H$="e" THEN
     LET E=EOF-1: LET FG2=1
5536 IF FG1<>1 AND G$>="A" THEN
     GO SUB 3000: GO TO 5500
5538 IF FG2<>1 AND H$>="A" THEN
     GO SUB 3000: GO TO 5500
5540 IF G$<"1" OR H$<"1" THEN
     GO SUB 3000: GO TO 5500
5542 LET A=VAL G$: LET E=VAL H$
5544 IF A<1 OR A>E OR E>=EOF
     THEN GO SUB 3000: GO TO
     5500
5550 FOR K=A TO E
5555    GO SUB 4100:
     IF FGP=0 THEN PAUSE 0
5560 IF INKEY$=" " THEN
     GO TO 5000
5565 IF INKEY$="M" THEN PRINT #1
     ;"INTERRUPT": PAUSE 0:
     GO TO 5565
5570 NEXT K
5575 LET FGP=0
5580 GO TO 5000
5590
5600 REM SEARCH
5601
5605 IF EOF=1 THEN GO TO 5005
5610 INPUT "AUTHOR(S) TITLE SOU"
     ;"RCE SIGNATURE"'"SPECIFY";
     " >"; LINE Z$
5612 IF Z$="EXIT" THEN GO TO 5E3
5614 IF LEN Z$<3 THEN GO SUB
     3200: GO TO 5610
5622 IF Z$(1 TO 3)="AUT" THEN
     LET JUMP=2000: GO TO 5630
5624 IF Z$(1 TO 3)="TIT" THEN
     LET JUMP=2100: GO TO 5630
5626 IF Z$(1 TO 3)="SOU" THEN
     LET JUMP=2200: GO TO 5630
5628 IF Z$(1 TO 3)="SIG" THEN
     LET JUMP=2300: GO TO 5630
5629 GO SUB 3000: GO TO 5610
5630
5632 INPUT "ENTER ITEM TO SEARC"
     ;"H FOR"'">"; LINE Z$
5634 IF Z$="" THEN GO SUB 3000:
     GO TO 5632
5638 INPUT TAB 9;"START, END, M"
     ;"AX. HITS?"'"SPECIFY >";
     LINE G$;TAB 16; LINE H$;TAB
     21; LINE F$
5639 INPUT "PRINTER OUTPUT ? >";
     LINE P$: IF P$="Y" OR P$=
     "y" THEN LET  FGP=1
5640 IF F$>="A" THEN GO SUB 3000
     : GO TO 5638
5642 IF F$="" THEN GO SUB 3000:
     GO TO 5610
5643 LET MAX=VAL F$
5644 LET FG1=0: LET FG2=0
5645 LET M=0
5646 IF G$="A" OR G$="a" THEN
     LET A=1: LET FG1=1
5647 IF H$="E" OR H$="e" THEN
     LET E=EOF-1: LET FG2=1
5648 IF FG1<>1 AND G$>="A" THEN
     GO SUB 3000: GO TO 5638
5650 IF FG2<>1 AND H$>="A" THEN
     GO SUB 3000: GO TO 5638
5652 IF G$<"1" OR H$<"1" THEN
     GO SUB 3000: GO TO 5638
5654 LET A=VAL G$: LET E=VAL H$
5656 IF A<1 OR A>E OR E>=EOF
     THEN GO SUB 3000: GO TO
     5638
5658
```

```
5660 LET X$=Z$
5661 LET A=A+1: LET E=E+1
5662 GO SUB JUMP
5665 PRINT #1; FLASH 1;"READY ";
     "SEARCHING WITH ";M;" HIT("
     ;"S)": PAUSE 0
5666 IF FGP=1 THEN LPRINT :
     PRINT #3;"READY SEARCHING "
     ;"WITH ";M;" HIT(S)": PAUSE
     0
5670 LET FGP=0: GO TO 5000
5675
5680 LET K=K-1: GO SUB 4100
5681 LET K=K+1
5684 IF FGP=0 THEN PRINT #1;"HI"
     ;"T ANY KEY TO CONTINUE SE"
     ;"ARCHING": PAUSE 0
5685 LET M=M+1: LET I=T-L
5690 IF M>=MAX THEN LET K=EOF
5695 RETURN
5699
5700 REM ENTER
5701
5710 GO SUB 4000
5720
5730 GO SUB 3700
5740
5750 GO TO 6000
5760

5800 REM ERASE
5801
5805 IF EOF=1 THEN GO TO 5005
5810 INPUT "ERASE PER NO./SIGNA"
     ;"TURE?"'"SPECIFY >"; LINE
     Z$
5815 IF Z$="EXIT" THEN GO TO 5E3
5820 IF Z$="NO" OR Z$="NO." THEN
     GO TO 5830
5821 IF LEN Z$<3 THEN GO TO 5824
5822 IF Z$(1 TO 3)="SIG" THEN
     GO TO 5850
5824 GO SUB 3200: GO TO 5800
5826
5830 INPUT "SPECIFY NO. TO ERAS"
     ;"E >"; LINE Z$
5831 IF Z$="EXIT" THEN GO TO 5E3
5833 LET K=VAL Z$
5834 IF K<1 OR K>=EOF THEN
     GO SUB 3010: GO TO 5830
5835 GO SUB 4100
5836 INPUT FLASH 1;"CONFIRM ERA"
     ;"SE: Y/N? >"; LINE Z$
5838 IF Z$<>"Y" AND Z$<>"y" THEN
     CLS : GO TO 5000
5839 LET EOF=EOF-1
5840 FOR I=K+1 TO EOF+1
5842   LET A$(I)=A$(I+1)
5843   LET B$(I)=B$(I+1)
5844   LET C$(I)=C$(I+1)
5845   LET D$(I)=D$(I+1)
5846 CLS : PRINT "WORKING":
     PAUSE 20: CLS : NEXT I
5847 PRINT #1;"O.K.-NO.";K;" ER"
     ;"ASED": PAUSE 200: CLS
5848 GO TO 5000
5849
5850 INPUT "ENTER SIGNATURE >";
     LINE Z$
5852 IF Z$="EXIT" THEN GO TO 5E3
5854 IF Z$="" THEN GO SUB 3200:
     GO TO 5850
5855 LET X$=Z$
5856 FOR K=2 TO EOF
5858   LET D=LEN D$(K)
5860   LET X=LEN X$
5862   LET L=X-1
5864   FOR I=1 TO D-L
5866     IF X$=D$(K,I TO I+L)
         THEN LET K=K-1: GO TO
         5835
5868   NEXT I
5869 CLS : PRINT "WORKING":
     PAUSE 20: CLS
5870 NEXT K
5875 CLS
5880 PRINT #1; FLASH 1;"NEGATIV"
     ;"E ERASE - NO SUCH ITEM"'
     "WITH SIGNATURE:"; FLASH 0;
     X$: PAUSE 0: GO TO 5000
5890
5900 REM CORRECT
5901
5905 IF EOF=1 THEN GO TO 5005
5910 INPUT "NO. TO CORRECT? >";
     LINE Z$
5912 IF Z$<"1" OR Z$>"99" THEN
     GO SUB 3000: GO TO 5900
5914 LET N=VAL Z$: IF N>=EOF
     THEN GO SUB 3E3: GO TO 5900
5920
5925 LET N=N+1
5930 LET A$(1)=A$(N)
5931 LET B$(1)=B$(N)
5932 LET C$(1)=C$(N)
5933 LET D$(1)=D$(N)
5940 GO SUB 3700
5950 LET A$(N)=A$(1)
5951 LET B$(N)=B$(1)
5952 LET C$(N)=C$(1)
5953 LET D$(N)=D$(1)
5960 CLS : GO TO 5000
5970

6000 REM INSERT
6001
6010 FOR I=2 TO EOF
6020   IF A$(1)>=A$(I) THEN NEXT
       I
6030 LET I0=I: LET EOF=EOF+1
6040 FOR I=EOF TO I0+1 STEP -1
6050   LET A$(I)=A$(I-1)
6052   LET B$(I)=B$(I-1)
6054   LET C$(I)=C$(I-1)
6056   LET D$(I)=D$(I-1)
6060 NEXT I
6070 LET A$(I0)=A$(1)
6072 LET B$(I0)=B$(1)
6074 LET C$(I0)=C$(1)
6076 LET D$(I0)=D$(1)
6080 LET K=I0-1: GO SUB 4100
6090 GO TO 5000
6100

9001 REM SAVE FILE
9010 CLS
9015 INPUT "CHANGE FILE NAME? Y"
     ;"/N? >"; LINE Z$: IF Z$<>
     "Y" AND Z$<>"y" THEN GO TO
     9030
9020 INPUT "ENTER NAME FOR DATA"
     ;" FILE >"'TAB 20; LINE N$
9030 LPRINT "FILE NAME = ";N$
9040 BEEP 1,18: SAVE N$ DATA
     A$()
9050 BEEP 1,18: SAVE N$ DATA
     B$()
9060 BEEP 1,18: SAVE N$ DATA
     C$()
9070 BEEP 1,18: SAVE N$ DATA
     D$()
9080 LET E(1)=EOF
9090 BEEP 1,18: SAVE N$ DATA E()
9099
9100 REM VERIFYING
9110 CLS
9130 PRINT #1; FLASH 1;"REWIND";
     " TAPE FOR VERIFYING DATA":
     PAUSE 0
9140 BEEP 1,18: VERIFY N$ DATA
     A$()
9150 BEEP 1,18: VERIFY N$ DATA
     B$()
9160 BEEP 1,18: VERIFY N$ DATA
     C$()
9170 BEEP 1,18: VERIFY N$ DATA
     D$()
9190 BEEP 1,18: VERIFY N$ DATA
     E()
9200 PRINT #1; FLASH 1;"POSITIV"
     ;"E VERIFYING": PAUSE 0
9300 STOP : STOP : STOP : STOP
9900
9990 CLEAR : SAVE "DATABASE"
     LINE 100
```

PIXEL für den ZX-Spectrum

Dieses Programm ermöglicht das Verschieben von definierten Zeichen um einen Bildschirmpunkt

Welcher Computerfan hat sich nicht schon über fehlende Programmier-Möglichkeiten seines Computers geärgert. Dieses Programm soll einen dieser Mängel beim ZX-Spectrum beseitigen.
Wie sicherlich jedem ZX-Spectrum-Besitzer bekannt ist, hat dieser Computer die Möglichkeit im hochauflösen Grafikmodus zu arbeiten. Jedoch können die definierten Zeichen immer nur um eine Print-Position verschoben werden. Auf der Suche nach einer Lösung diese Zeichen um jeweils einen Bildschirmpunkt zu verschieben, stießen wir im ROM dieses Computers auf eine Routine bei der Adresse $22E5 (8933dez). Diese Routine erwartet im Register B die Y-Koordinate (0–175) und im Register C die X-Koordinate (0–255). Wird diese Routine aufgerufen, so setzt der Computer einen Punkt. Mit Hilfe dieser Routine wurde folgendes Maschinenprogramm geschrieben, welches es ermöglicht, Zeichen auf den Bildschirm zu plotten. Da das Programm mit OVER 1 und 0 arbeitet, ist es möglich, Zeichen ineinander zu drukken. Außerdem können beliebig viele Grafikzeichen definiert werden, da der Zeichenbereich an jede beliebige Adresse gelegt werden kann. Die unterstrichenen Bytes im Assemblerlisting stellen Daten dar.

8001 hex Bei einer 1 wird OVER 1 gesetzt, bei einer 0 OVER 0

8011 hex; 8012 hex Ist die Adresse des Zeichens, jedoch muß das LOW-Byte bei 8011 eingegeben werden. In der vorliegenden Form wird das Zeichen mit dem Code 64 ausgegeben. Um beispielsweise B auf den Bildschirm zu plotten, müssen Sie 8011 hex mit 16 dez (10 hex) laden.

8014 hex; 8015 hex Dies sind X- bzw. Y-Koordinaten. Wird der zulässige Zahlenbereich über- bzw. unterschritten, so erfolgt ein INTEGER OUT OF RANGE ERROR.

Da die Routine frei verschiebbar ist, müssen bei einer Änderung der Startadresse die Datenadressen natürlich mit angepaßt werden. Das Programm läuft je nach gewählter Startadresse auf dem 16- als auch 48k Spectrum.

Assemblerlisting!

```
        8000  3e 01          ld a (data over)
          02  fe 01          cp a 1
          04  28 06          jr nz over 1
over0     06  fd cb 57 86     res 0 (iy+87)
          0a  18 04          jr ldata
over1     0c  fd cb 57 c6     set 0 (iy+87)
ldata     10  21 00 3e       ld hl (data adr)
          13  01 80 50       ld bc (data coord)
          16  16 08          ld d 8
nbyte     18  7e             ld a (hl)
          19  e5             push hl
          1a  d5             push de
          1b  16 07          ld d 7
nebit     1d  d5             push de
          1e  cb 7f          bit 7 a
          20  28 07          jr nz nePos
Point     22  e5             push af
          23  c5             push bc
          24  cd e5 22        call 22e5
          27  c1             pop bc
          28  f1             pop af
nePos     29  0c             inc c
          2a  cb 17          rl a
          2c  d1             pop de
          2d  15             dec d
          2e  20 ed          jr nz nebit
          30  79             ld a c
          31  d6 07          sub a 7
          33  4f             ld c a
          34  05             dec b
          35  d1             pop de
          36  e1             pop hl
          37  23             inc hl
          38  15             dec d
          39  20 dd          jr nz nbyte
          3b  c9             ret
```

Basic-Loader!

```
   2 LET m$="3e01fe002806fdcb5786180 4fdcb57c621003e01805016087ee5d51607d
5cb7f2807f5c5cde522c1f10ccb17d11520ed79d6074f05d1e1231520ddc9"
  20 LET m$=m$+"*"
  25 LET z=0
  30 LET a=32768
  40 LET p=16*(CODE m$-48-(39*(m$>"9"))+CODE m$(2)-48-(39*(m$(2)>"9"))
  50 POKE a,p
  55 LET z=z+p
  60 LET m$=m$(3 TO )
  70 IF m$="*" THEN  GO TO 100
  80 LET a=a+1
  90 GO TO 40
 100 IF z<>6781 THEN  PRINT "Fehler im M-CODE!!"
 110 STOP
```

Spritehilfe für C-64

Eine wesentliche Erweiterung für den C-64 mit über 20 Funktionen

Um auch Anfängern die Arbeit mit dieser Befehlserweiterung zu ermöglichen, müssen einige Grundlagen geklärt werden.

Block
Ein Block ist der Inhalt eines bestimmten Speicherplatzes im Computer, in dem das Aussehen eines Sprites definiert ist. Für einen Spriteblock werden 64 Bytes reserviert, von denen jedoch nur 63 Bytes gebraucht werden.
Die Adresse eines Blocks errechnet sich aus der Basisadresse des VICs (Video Interface Controler) + Blocknummer * 64.
Der Bereich für eine Blocknummer erstreckt sich in dieser Befehlserweiterung von 128 bis 255.
Eine Blocknummer, die kleiner ist als 128, würde die Spritehilfe überschreiben, also zerstören. Deshalb wird bei Unterschreitung dieses Wertes die Fehlermeldung 'WRONG BLOCKNUMMER ERROR' ausgegeben.

Animation
Unter Animation versteht man bestimmte Bewegungsabläufe. Eine Animation ist zum Beispiel jeder Zeichentrickfilm. Bei einer Animation wird sehr schnell von einem Bild auf ein anderes umgeschaltet, so daß bei entsprechenden Bildern der Eindruck einer Bewegung entstehen kann.
Eine Animation kann auch mit Sprites realisiert werden, wozu man verschiedene Spritedefinitionen hintereinander im Rechner ablegt und dann der Reihe nach, immer auf die nächste Definition umschaltet. Dadurch kann eine Figur richtig laufen lernen oder die Rotorblätter eines Hubschraubers bewegt werden (siehe Animationsdemo).
Das erzielt besonders in Spielen einen sehr guten grafischen Effekt.

Interruptprogrammierung
Eine sehr gute Eigenschaft des Commodore 64 ist die sogenannte Interruptrequest (IRQ). Ein IRQ wird nach Ablauf einer einstellbaren Zeitspanne ausgelöst. Das veranlaßt den 6510 Prozessor, den Assemblerbefehl (nicht Basic-Befehl), der gerade abgearbeitet wird, noch auszuführen, dann in eine Routine zu springen und später wieder in das eigentliche Programm zurückzukehren. Diese IRQ-Routine läßt beispielsweise den Cursor blinken, oder die Tastatur abfragen usw.
Man kann aber auch eine eigene IRQ-Routine schreiben. Dieses wurde in dieser Spritehilfe realisiert. In der neuen IRQ-Routine können jetzt Sprites, unabhängig vom weiteren Programmablauf, bewegt oder animiert werden.

Erklärung der Befehle im einzelnen:

S-ON sp# (,sp#,sp#3,...)
Sprites können durch diesen Befehl eingeschaltet werden. Bei Überschreitung einer Spritenummer (>7) wird die Fehlermeldung 'WRONG SPRITENUMBER ERROR' ausgegeben. Es können beliebig viele Spritenummern durch Kommata getrennt folgen. Das gilt auch für alle folgenden Befehle, deren Parameter in Klammern stehen.

S-OFF sp# (,sp#2,...)
Dieser Befehl bildet das Gegenstück zu S-ON. Hiermit werden die Sprites wieder ausgeschaltet.

SSET sp#,block,col,Prior (,sp#2,block2,col2,Prior2 ,...)
Dieser Befehl beinhaltet drei verschiedene Funktionen. Zum ersten wird die Blocknummer (block) bestimmt, aus der die Definition für das Sprite NR. sp# stammt. Zum zweiten wird die Farbe (col) für das Sprite sp# festgelegt (0–15) und als letztes wird die Priorität eines Sprite bestimmt. Wird für die Priorität der Wert Null angegeben, so erscheint das Sprite vor dem Hintergrund, bei Prior=1 erscheint es hinter dem Hintergrund.

S-EXPAND sp#,exp
Ein Sprite kann in zwei Richtungen vergrößert werden. Wenn exp=0 ist, dehnt sich das Sprite in Y-Richtung aus, ist exp=1 dehnt es sich in X-Richtung aus, bei exp=2 dehnt es sich in beide Richtungen aus, und wenn exp=3 ist erhält das Sprite wieder seine Normalgröße.

SPOSIT sp#,X,Y
Mit diesem Befehl kann ein Sprite positioniert werden. Es erscheint nach Eingabe des Befehls an den Koordinaten X (0–511) und Y (0–255). Ein Sprite kann auch aus dem Bildschirm herausbewegt werden. Das ist der Fall, wenn die Koordinaten kleiner bzw. größer als die folgenden Werte sind.

X-Richtung : kleiner als 40 ; größer als 340.
Y-Richtung : kleiner als 50 ; größer als 210.

Die Koordinate, an der das Sprite den Bildschirm verläßt, ist ganz vom Aussehen und der Größe eines Sprites abhängig. Folglich sind die o. g. Werte auch nur sehr ungenau.
Zum Schluß sei noch gesagt, daß dieser Befehl ein Sprite ca. 3mal so schnell positioniert, wie der herkömmliche POKE-Befehl.

Es folgen die Befehle, die ein Sprite unabhängig vom Programmablauf steuern:

SET TIMER tn
SET TIMER initialisiert die neue IRQ-Routine. Das muß vor dem Arbeiten mit den Befehlen SMOVE und SANIMATE geschehen. Nach Eingabe des Befehls läuft ständig ein Countdown der Variablen tn ab. Ist dieser abgelaufen, wird ein IRQ ausgelöst und in eine Routine gesprungen, wo, wenn es erforderlich ist, Sprites bewegt werden. Nach Ablauf des Timers startet dieser wieder selbständig.

TIMER OFF
Der Befehl TIMER OFF schaltet den Timer wieder aus. Es erfolgt ein Stillstand aller IRQ-gesteuerten Sprites.

SMOVE sp#,X,Y,RX,RY,SX,SY
Durch diesen Befehl kann nun ein Sprite unabhängig gesteuert werden.
Die Parameter sind im folgenden erklärt:

sp#: Spritenummer
X: X-Startkoordinate
Y: Y-Startkoordinate
RX: Richtung einer Zeile. Wenn RX=0 ist, bewegt sich das Sprite nach rechts, ist RX=1 bewegt es sich nach links.
RY: Richtung einer Spalte. Bei RY=0 bewegt sich das Sprite nach unten, ist RX=1 bewegt es sich nach oben.
SX: Zeilengeschwindigkeit eines Sprites (0=Sprite steht; 1=schnell; 255=langsam)
SY: Spaltengeschwindigkeit eines Sprites.

SANIMATE sp#,X,Y,RX,RY,SX,SY, block,sw,a-speed
SANIMATE animiert ein Sprite IRQ-gesteuert. Die Parameter entsprechen bis zur Variablen SY dem Befehl SMOVE. Zusätzlich werden noch die Parameter für die Animation verlangt, die im folgenden erklärt sind.

block: Die Animation wird ab der Blocknummer "block" gestartet.
sw: Schrittweite einer Animation. Mit sw wird die Anzahl der Spritedefinitionen angegeben die bei einer Animation durchlaufen werden.
a-speed: Mit a-speed schließlich wird die Geschwindigkeit angegeben, mit der die Spritedefinitionen umgeschaltet werden.

S-STOP sp# (,sp#2,...)
S-STOP hält ein Sprite, das durch SANIMATE oder SMOVE in Bewegung gesetzt wurde, an.

Um nicht immer die meterlangen Parameterwüsten der Befehle SMOVE und SANIMATE anzugeben, habe ich noch einige Teilbefehle geschrieben.

SSPEED sp#,SX,SY
Dieser Befehl ändert die Geschwindigkeit eines Sprites.

SCOURSE sp#,RX,RY
Durch diesen Befehl kann die Richtung eines Sprites geändert werden.

SREMOVE sp#,SX,SY,RX,RY
Durch SREMOVE kann ein Sprite von seinem Standpunkt aus bewegt werden. D. h., daß man die Startkoordinaten nicht anzugeben braucht.

SREANIM sp#,block,sw,a-speed
Einem Sprite wird durch SREANIM eine neue Blocklage sowie eine neue Länge und Geschwindigkeit für eine Animation zugeteilt.

Folgende Befehle bestimmen das Aussehen eines Sprites

DES BEGIN block
Durch DES BEGIN wird die Blocknummer festgelegt, an der das Sprite definiert wird.

SD
Durch SD wird das Aussehen eines Sprite festgelegt. Als Beispiel siehe Animationsdemo.

DES END
Damit nicht alle 21 Zeilen eines Sprites angegeben werden müssen, kann ein vorzeitiger Abbruch mit DES END erfolgen.

Funktionen:

BLOCK (block)
Diese Funktion ergibt die Adresse eines Blocks (entspricht PRINT 49152+block*64).

BKOL
Durch BKOL kann der Zusammenstoß eines oder mehrerer Sprites registriert werden. Aus dem Bitmuster, das sich ergibt, können die Spritenummern ermittelt werden.

SKOL
Eine Sprite-Sprite Kollision kann durch die Funktion SKOL ermittelt werden. Funktionsweise wie BKOL.

VIC
Die Befehlsfolge "PRINT VIC" ergibt immer die Adresse des VICs (53248). So braucht diese Basisadresse nicht durch eine Variable definiert werden, die 6 Bytes mehr Speicherplatz benötigt.

SCRN
Die Funktion SCRN ergibt den Wert 52224. Dieser Wert ist der Anfang des verschobenen Bildschirm-Rams (sonst 1024).
Die Befehlsfolge POKE SCRN,1:POKE 55296,5 setzt in der linken oberen Ecke des Bildschirms ein grünes A.

Da ich nach Programmierung der Spritebefehle noch Speicherplatz zur Verfügung hatte, habe ich ein kleines Toolkit in die Spritehilfe integriert.

COLOR ,pf
Setzen der Punktfarbe geschieht durch diesen Befehl.

COLOR rf,hf (,pf)
Durch diese Variante des COLOR-Befehls wird die Rahmenfarbe und die Hintergrundfarbe des Bildschirms bestimmt. Die Punktfarbe kann, muß aber nicht angegeben werden.

DISPLAY X,Y (,"string")
DISPLAY setzt, sofern kein String angegeben ist, den Cursor an die Position X und Y. Wenn ein String mit angegeben ist wird jener an diese Stelle geschrieben.

WALK modus
Durch den Befehl WALK wird das Austesten von Basic-Programmen enorm erleichtert. Dieser Befehl löst einen sogenannten Einzelschrittmodus aus. WALK 0 schaltet den Einzelschrittmodus ein. Wenn man jetzt ein Programm startet, wird als erstes die laufende Zeilennummer in der rechten oberen Ecke angezeigt. Dann wartet das Programm solange auf die Abarbeitung des nächsten Befehls, bis man die SHIFT,C= oder CTRL-Taste gedrückt hat.
Der Einzelschrittmodus kann mit WALK 1 wieder ausgeschaltet werden. Eine Unterbrechung eines Programms im Einzelschrittmodus ist nur mit STOP/SHIFT möglich.

DISK "Diskettenbefehl"
Der Befehl DISK sendet einen Befehl an das Diskettenlaufwerk.
Beispiel:
DISK "N:TESTDISK,ID" formatiert eine Diskette neu.

CATALOG ("$:anweisung")
Der Befehl CATALOG zeigt den Inhalt einer Diskette ohne ein Basicprogramm zu zerstören. Wird zusätzlich noch ein String angegeben, so ist eine sortierte Ausgabe des Diskketteninhaltes möglich.
Beispiel:
CATALOG: Listet den Inhalt einer Diskette; CATALOG "$:A*": Listet nur die Programme die mit einem A anfangen; CATALOG "$:*=SEQ":Listet nur Sequenzielle Files.

RESET
Dieser Befehl schaltet die Spritehilfe aus und springt zum Kaltstart des C-64. Ein Aufruf der Spritehilfe ist mit SYS 49152 möglich.

Fehlermeldungen:

S-DEFINITION TO BIG
Es wurde versucht ein Sprite mit mehr als 21 Zeilen zu definieren. Einer Spritedefinition muß immer der Befehl DES BEGIN vorangehen.

TO FEW ARGUMENTES
In einem SD Befehl wurden weniger als 24 Zeichen angegeben.

TO MANY ARGUMENTES
In einem SD Befehl wurden mehr als 24 Zeichen angegeben.

WRONG BLOCKNUMBER
Es wurde versucht ein Block anzusprechen, der kleiner als 128 war.

WRONG SPRITENUMBER
Es wurde versucht eine Spritenummer größer als 7 anzusprechen. (Es gibt 8 mögliche Sprite, durchnumeriert von 0–7).

TIMENUMBER TO LOW
Diese Fehlermeldung wird ausgegeben, wenn der Timer durch den Befehl SET TIMER, mit einem kleinerem Wert als 2048 geladen werden soll. Kleinere Werte wären gefährlich, da das Programm in eine Endlosschleife beim Abarbeiten der IRQ-Routine geraten könnte. Es würde sofort abstürzen.

TO MANY CARACTERS
Bei dem DISK Befehl wurde ein String mit mehr als 40 Zeichen angegeben. Das Diskettenlaufwerk kann nur Strings bearbeiten, deren Länge kleiner als 41 ist.

Wichtige Hinweise zur Anwendung des Programms

1) Der Bildschirmspeicher ist verschoben. Die jetzige Adresse ist 52224 (sonst 1024).
2) Die Spritepointer (sonst 2040–2047) sind nach 53240–53247 verschoben.
3) Vor Bedienung der Peripherie, muß der Timer unbedingt ausgeschaltet werden, da es sonst zu einem Absturz kommen kann. Der Befehl CATALOG schaltet den Timer selbständig ab.
4) Es ist kein RUN/STOP-RESTORE Schutz mit POKE 792,193 möglich, da der MINI-Vektor 'verbogen' ist.

```
90 POKE53280,0:POKE53281,0:GOTO250
99 :
100 ********************************
110 ***                          ***
120 ***                          ***
130 ***       SPRITEHILFE        ***
140 ***                          ***
150 ***   COPYRIGHT 1984 BY      ***
160 ***                          ***
170 ***   KLAUS LOEFFELMANN      ***
180 ***                          ***
200 ***   4760 LIPPSTADT 16      ***
210 ***                          ***
220 ***                          ***
230 ********************************
240 :
250 DIM ZN(20),PS(20)
260 :
270 PRINT"□SPRITEHILFE -- BASIC-LOADER --"
280 FOR Z=1TO20:READ ZN(Z):NEXT
290 FOR Z=0TO20:READ PS(Z):NEXT
300 :
310 PRINT"■■ML-CODE WIRD GELESEN■"
320 :
330 FOR Z1=1 TO 20
340 PRINT Z1;:IFZ1<10THENPRINT"■.  ";
345 IFZ1>9THENPRINT"■. ";
346 PRINT "SEQUENZ";
350 FOR ZZ=1 TO 154
360 READ P:POKE49152+ZZ,P:ZZ=ZZ+1
370 PS=PS+P:NEXT ZZ
380 IF PS<>PS(Z1)THEN450
390 PRINT "- KEIN FEHLER":PS=0
400 NEXT Z1
410 :
420 PRINT"□"
430 PRINT "MIT 'SYS 49152' STARTEN"
```

```
440 END
445 :
450 REM FEHLER
460 :
470 PRINT "FEHLER IN DATAS ZEILE:";PEEK(63)+256*PEEK(64)
480 END
490 :

500 REM ZEILNUMMERN
510 :
520 DATA 1000,1018,1025,1033,1042,1050
530 DATA 1058,1066,1074,1082,1090,1099
540 DATA 1107,1115,1123,1132,1140,1148
550 DATA 1156,1164,1172
560 :
570 REM PRUEFSUMMEN
580 :
590 DATA 10205,15847,20569,21802,17259

600 DATA 18974,19343,20176,19773,19826
610 DATA 20001,19662,20401,20458,21469
620 DATA 20590,21449,15933,18277,17033
630 :

1000 REM MC-DATAS
1010 :
1011 DATA169,11,141,17,208,76,192,194,147,30,42,42,42,42,32,32,32,67,66,77
1012 DATA45,54,52,32,69,68,84,69,78,68,69,68,32,66,65,83,73,67,32,86,49,46
1013 DATA48,32,32,32,42,42,42,42,67,79,80,89,82,73,71,72,84,32,32,40,67,41
1014 DATA32,49,57,56,52,32,66,69,32,75,76,65,85,83,32,76,79,69,70,70,69,76
1015 DATA77,65,78,78,45,45,45,45,45,45,45,45,32,51,56,57,49,49,32,66,65,83
1016 DATA73,67,45,66,89,84,69,83,32,70,82,69,69,32,45,45,45,45,45,45,45,45
1017 DATA13,83,45,79,206,83,83,80,69,69,196,83,69,84,32,84,73,77,69,210,83
1018 DATA45,69,88,80,65,78,196,83,67,79,85,82,83,197,83,45,83,84,79,208,83
1019 DATA45,79,70,198,84,73,77,69,82,32,79,70,198,83,83,69,212,83,80,79,83

1020 DATA73,212,83,65,78,73,77,65,84,197,83,77,79,86,197,83,82,69,77,79,86
1021 DATA197,83,82,69,65,78,73,205,67,79,76,79,210,68,73,83,80,76,65,217,82
1022 DATA69,83,69,212,67,65,84,65,76,79,199,68,69,83,32,66,69,71,73,206,83
1023 DATA196,68,69,83,32,69,78,196,68,73,83,203,87,65,76,203,66,76,79,67,203
1024 DATA66,75,79,204,83,75,79,204,86,73,195,83,67,82,206,0,236,196,230,197
1025 DATA10,198,59,198,198,198,242,198,5,197,0,199,14,199,38,197,114,199,137
1026 DATA197,185,199,191,199,75,200,107,200,157,200,165,200,97,201,193,201
1027 DATA160,201,7,203,132,203,198,199,39,200,51,200,109,203,121,203,166,122
1028 DATA160,4,132,15,189,0,2,16,7,201,255,240,62,232,208,244,201,32,240,55
1029 DATA133,8,201,34,240,85,36,15,112,45,201,63,208,4,169,153,208,37,201

1030 DATA48,144,4,201,60,144,29,132,113,160,0,132,11,136,134,122,202,200,232
1031 DATA189,0,2,56,249,158,160,240,245,201,128,208,47,5,11,164,113,232,200
1032 DATA153,251,1,201,0,240,56,56,233,58,240,4,201,73,208,2,133,15,56,233
1033 DATA85,208,160,133,8,189,0,2,240,224,197,8,240,220,200,153,251,1,232
1034 DATA208,240,166,122,230,11,200,185,157,160,16,250,185,158,160,208,181
1035 DATA240,15,169,0,2,16,169,153,253,1,198,123,169,255,133,122,96,160,0
1036 DATA185,131,192,208,2,200,232,189,0,2,56,249,131,192,240,245,201,128
1037 DATA208,4,5,11,208,153,166,122,230,11,200,185,130,192,16,250,185,131
1038 DATA192,208,224,240,198,16,15,36,15,48,11,201,255,240,7,201,204,176,6
1039 DATA76,36,167,76,243,166,56,233,203,170,132,73,160,255,202,240,8,200

1040 DATA185,131,192,16,250,48,245,200,185,131,192,48,5,32,71,171,208,245
1041 DATA76,239,166,32,152,203,32,115,0,32,97,194,76,174,167,201,204,144,4
1042 DATA201,227,144,6,32,121,0,76,237,167,56,233,204,10,170,189,38,193,72
1043 DATA189,37,193,72,76,115,0,169,0,133,13,32,115,0,201,227,144,4,201,232
1044 DATA144,6,32,121,0,76,141,174,201,227,208,14,56,233,227,10,72,32,115
1045 DATA0,32,241,174,76,174,194,56,233,227,10,72,32,115,0,104,170,189,83
1046 DATA193,133,85,189,84,193,133,86,32,84,0,76,141,173,169,0,141,32,208
1047 DATA141,33,208,32,68,166,120,216,32,163,253,32,21,253,32,27,229,32,152
1048 DATA195,169,0,162,17,157,255,207,202,208,250,169,93,160,193,141,4,3,140
```

```
1049 DATA5,3,169,32,160,194,141,6,3,140,7,3,169,85,160,194,141,8,3,140,9,3

1050 DATA169,127,160,194,141,10,3,140,11,3,162,0,189,8,192,32,210,255,232
1051 DATA224,123,208,245,162,9,189,255,127,157,99,203,202,208,247,169,0,160
1052 DATA192,141,0,128,140,1,128,169,0,141,21,208,141,14,221,169,206,141,24
1053 DATA3,160,195,140,25,3,162,5,189,15,253,157,3,128,202,208,247,32,133
1054 DATA200,169,27,141,17,208,76,134,227,120,169,55,133,1,169,0,141,14,221
1055 DATA169,127,141,13,221,172,13,221,48,44,32,188,246,32,225,255,208,36
1056 DATA32,27,229,169,5,141,134,2,169,0,141,32,208,141,33,208,169,0,141,21
1057 DATA208,141,14,221,141,59,198,141,23,208,141,29,208,108,2,160,76,114
1058 DATA254,120,160,0,132,250,169,208,133,251,162,16,169,51,133,1,177,250
1059 DATA72,169,48,133,1,104,145,250,200,208,239,230,251,202,208,234,169,55

1060 DATA133,1,169,52,141,24,208,169,0,141,0,221,169,204,141,136,2,88,96,72
1061 DATA138,72,152,72,172,13,221,152,41,1,208,3,76,85,195,76,24,196,0,0,0
1062 DATA142,225,195,138,10,141,226,195,174,225,195,189,16,196,141,227,195
1063 DATA160,0,174,226,195,173,16,208,77,227,195,141,16,208,152,157,0,208
1064 DATA174,225,195,157,0,160,76,39,196,1,2,4,8,16,32,64,128,169,54,133,1

1065 DATA162,0,189,8,160,201,1,240,56,176,15,232,224,8,208,242,169,55,133
1066 DATA1,104,168,104,170,104,64,222,88,160,208,34,189,80,160,157,88,160
1067 DATA254,248,207,189,248,207,221,72,160,208,17,189,48,160,157,248,207
1068 DATA76,93,196,0,2,4,6,8,10,12,14,189,40,160,240,52,189,64,160,208,25
1069 DATA222,32,160,208,42,189,40,160,157,32,160,138,72,189,85,196,170,254

1070 DATA1,208,104,170,76,150,196,222,32,160,208,17,189,40,160,157,32,160
1071 DATA138,72,189,85,196,170,222,1,208,104,170,189,24,160,240,68,189,56
1072 DATA160,208,28,222,16,160,208,58,189,24,160,157,16,160,254,0,160,32,57
1073 DATA201,169,0,160,205,56,201,144,22,76,228,195,222,16,160,208,30,189
1074 DATA24,160,157,16,160,222,0,160,208,3,76,228,199,189,0,160,168,138,72
1075 DATA189,85,196,170,152,157,0,208,104,170,76,39,196,32,158,183,224,8,144
1076 DATA3,76,3,200,96,32,226,196,189,16,196,141,5,197,173,21,208,13,5,197
1077 DATA141,21,208,32,63,200,240,233,96,0,32,226,196,189,16,196,141,5,197
1078 DATA169,255,56,237,5,197,141,5,197,173,5,197,45,21,208,141,21,208,32
1079 DATA63,200,240,224,96,32,226,196,134,250,32,253,174,32,235,183,134,251

1080 DATA165,250,10,141,5,197,165,251,174,5,197,157,1,208,165,20,166,250,157
1081 DATA0,160,174,5,197,157,0,208,165,21,240,22,169,1,166,250,240,3,189,16
1082 DATA196,141,5,197,173,16,208,13,5,197,141,16,208,96,169,1,169,1,166,250
1083 DATA240,3,189,16,196,141,5,197,56,169,255,237,5,197,141,5,197,173,16
1084 DATA208,45,5,197,141,16,208,96,32,39,197,32,253,174,32,158,183,224,2
1085 DATA144,3,76,72,178,138,166,250,157,56,160,32,253,174,32,158,183,224
1086 DATA2,144,3,76,72,178,138,166,250,157,64,160,169,1,157,8,160,32,253,174
1087 DATA32,158,183,138,166,250,157,16,160,157,24,160,32,253,174,32,158,183
1088 DATA138,166,250,157,32,160,157,40,160,96,32,158,183,224,8,144,3,76,3
1089 DATA203,134,250,32,253,174,96,32,215,197,32,158,183,138,166,250,157,16

1090 DATA160,157,24,160,32,253,174,32,158,183,138,166,250,157,32,160,157,40
1091 DATA160,32,63,200,240,221,96,32,138,173,32,247,183,165,21,201,8,176,5
1092 DATA162,5,76,243,202,173,59,198,208,15,169,17,141,14,221,169,129,141
1093 DATA13,221,169,1,141,59,198,165,20,164,21,141,4,221,140,5,221,96,0,32
1094 DATA215,197,32,158,183,134,251,169,1,166,250,240,4,10,202,208,252,133
1095 DATA250,169,255,229,250,133,2,165,251,240,18,201,1,240,36,201,2,240,54
1096 DATA201,3,240,72,32,63,200,240,209,96,173,23,208,5,250,141,23,208,173
1097 DATA29,208,37,2,141,29,208,32,63,200,240,187,96,173,29,208,5,250,141
1098 DATA29,208,173,23,208,37,2,141,23,208,32,63,200,240,165,96,173,29,208
1099 DATA5,250,141,29,208,173,23,208,5,250,141,23,208,32,63,200,240,143,96

1100 DATA173,29,208,37,2,141,29,208,173,23,208,37,2,141,23,208,32,63,200,240
1101 DATA1,96,76,60,198,32,215,197,32,158,183,224,2,144,3,76,72,178,138,166
1102 DATA250,157,56,160,32,253,174,32,158,183,224,2,144,3,76,72,178,138,166
1103 DATA250,157,64,160,32,63,200,240,213,96,32,226,196,169,0,157,8,160,32
1104 DATA63,200,240,243,96,169,0,141,14,221,141,13,221,169,0,141,59,198,96
1105 DATA32,215,197,32,158,183,224,128,176,5,162,2,76,243,202,138,166,250
1106 DATA157,248,207,32,253,174,32,158,183,224,17,144,3,76,72,178,138,166
1107 DATA250,157,39,208,169,1,166,250,189,16,196,133,251,169,0,229,251,133
1108 DATA252,32,253,174,32,158,183,224,2,144,3,76,72,178,224,1,240,14,173
1109 DATA27,208,37,252,141,27,208,32,63,200,240,171,96,173,27,208,5,251,141
```

```
1110 DATA27,208,32,63,200,240,157,96,32,138,197,32,253,174,32,158,183,224
1111 DATA128,176,5,162,2,76,243,202,138,166,250,157,48,160,157,248,207,32
1112 DATA253,174,32,156,183,224,128,144,5,162,2,76,243,202,138,105,128,166
1113 DATA250,157,72,160,32,253,174,32,158,183,138,166,250,157,80,160,157,88
1114 DATA160,169,2,157,8,160,96,32,215,197,76,144,197,32,215,197,76,121,199
1115 DATA76,211,199,144,64,0,0,0,135,0,0,0,0,169,206,160,199,32,40,186,169
1116 DATA201,160,199,76,103,184,0,0,0,142,225,199,138,10,141,226,199,174,225
1117 DATA199,189,16,196,141,227,199,173,29,208,45,227,199,208,27,174,226,199
1118 DATA160,255,173,227,199,77,16,208,141,16,208,152,157,0,208,174,225,199
1119 DATA157,0,160,76,39,196,173,16,208,45,227,199,208,221,174,226,199,160
1120 DATA247,208,219,172,31,208,32,162,179,169,0,141,31,208,96,172,30,208
1121 DATA32,162,179,169,0,141,30,208,96,32,121,0,201,44,208,5,32,115,0,169
1122 DATA0,96,32,63,200,240,20,32,158,183,142,32,208,32,253,174,32,158,183
1123 DATA142,33,208,32,63,200,208,6,32,158,183,142,134,2,96,32,158,183,138
1124 DATA72,32,253,174,32,156,183,104,168,24,32,240,255,32,63,200,208,233
1125 DATA76,164,170,169,0,133,97,169,224,133,98,162,32,169,0,160,0,145,97
1126 DATA200,208,251,230,98,202,208,246,96,162,0,141,6,128,76,226,252,32,121
1127 DATA0,201,0,240,3,76,57,203,169,147,32,210,255,169,0,141,14,221,141,21
1128 DATA208,169,36,162,251,160,0,133,251,134,187,132,188,169,1,162,8,160
1129 DATA96,133,183,134,166,132,165,32,213,243,165,186,32,180,255,165,185
1130 DATA32,150,255,169,0,133,144,160,3,132,251,32,165,255,133,252,164,144
1131 DATA208,59,32,165,255,164,144,208,52,164,251,136,208,233,166,252,32,205
1132 DATA189,169,32,120,32,210,255,32,165,255,166,144,208,29,170,240,6,32
1133 DATA210,255,76,10,201,169,13,32,210,255,173,141,2,201,4,240,249,201,2
1134 DATA240,4,160,2,208,166,32,66,246,169,1,76,145,242,0,0,0,142,54,201,169
1135 DATA1,174,54,201,240,4,10,202,208,252,141,55,201,173,16,208,45,55,201
1136 DATA208,9,169,255,141,56,201,174,54,201,96,169,247,141,56,201,208,245
1137 DATA169,0,141,158,201,141,159,201,32,158,183,224,128,176,5,162,2,76,243
1138 DATA202,169,0,24,105,64,144,3,238,159,201,202,208,245,24,105,0,144,3
1139 DATA238,159,201,141,158,201,173,159,201,24,105,192,141,159,201,169,0

1140 DATA141,160,201,24,96,0,224,0,173,160,201,201,63,240,22,173,158,201,172
1141 DATA159,201,133,254,132,255,169,0,172,160,201,145,254,200,192,64,208
1142 DATA249,96,0,0,0,24,32,158,173,36,13,48,3,76,153,173,32,166,182,201,24
1143 DATA240,12,144,5,162,4,76,243,202,162,1,76,243,202,32,237,201,32,121
1144 DATA0,96,1,0,0,0,0,0,234,160,0,140,232,201,169,128,141,231,201,162,0
1145 DATA177,34,201,46,240,17,201,42,240,3,76,8,175,173,233,201,24,109,231
1146 DATA201,141,233,201,173,231,201,24,74,24,141,231,201,200,232,224,8,208
1147 DATA218,174,232,201,173,233,201,157,234,201,238,232,201,169,128,141,231
1148 DATA201,169,0,141,233,201,173,232,201,162,0,201,3,240,3,76,250,201,173
1149 DATA158,201,172,159,201,133,254,132,255,172,160,201,192,63,144,5,162
1150 DATA0,76,243,202,162,0,189,234,201,145,254,238,160,201,200,232,224,3
1151 DATA208,242,96,83,45,66,69,70,73,78,73,84,73,79,78,32,84,79,32,66,73
1152 DATA78,79,77,32,70,69,87,32,65,82,71,85,77,69,78,84,69,211,87,82,79
1153 DATA78,71,32,66,76,79,67,75,78,85,77,66,69,210,87,82,79,78,71,32,83,80
1154 DATA82,73,84,69,78,85,77,66,69,210,84,79,32,77,65,78,89,32,65,82,71,85
1155 DATA77,69,78,84,69,211,84,73,77,69,78,85,77,66,69,82,32,84,79,32,76,79
1156 DATA215,84,79,32,77,65,78,89,32,67,72,65,82,65,67,84,69,82,211,105,202
1157 DATA124,202,141,202,158,202,176,202,194,202,211,202,138,10,170,189,229
1158 DATA202,133,34,189,230,202,133,35,76,71,164,162,3,76,243,202,24,32,158
1159 DATA173,36,13,48,3,76,153,173,32,166,182,201,41,144,5,162,6,76,243,202

1160 DATA72,169,1,162,8,160,15,32,186,255,104,166,34,164,35,32,189,255,32
1161 DATA192,255,169,1,76,195,255,24,32,158,173,36,13,48,3,76,153,173,32,166
1162 DATA182,72,165,34,166,35,133,187,134,188,169,0,141,21,208,141,14,221
1163 DATA169,147,32,210,255,32,121,0,104,76,203,200,169,110,160,133,32,30
1164 DATA171,32,131,169,116,160,203,76,162,187,144,80,0,0,0,169,128,160,203
1165 DATA76,162,187,144,76,0,0,0,32,158,183,224,0,208,6,169,1,141,221,203
1166 DATA96,169,0,141,221,203,96,165,157,48,64,173,221,203,240,59,56,32,10
1167 DATA229,152,72,138,72,160,33,162,0,24,32,10,229,169,238,160,203,32,30
1168 DATA171,160,34,162,0,24,32,10,229,169,18,32,210,255,166,57,165,58,32
1169 DATA205,189,169,146,32,210,255,32,222,203,104,170,104,168,24,32,10,229
1170 DATA96,0,173,141,2,240,251,173,141,2,208,251,160,255,136,208,253,96,18
1171 DATA35,32,32,32,32,32,32,146,0,0,41,175,255,164,255
1172 DATA 0,0,0,0,0,0,0,0,0,0
```

Animationsdemo:

```
1 REM ************************
2 REM **  ANIMATIONSDEMO    **
3 REM **          BY          **
4 REM ** KLAUS LUEFFELMANN **
5 REM ************************
6 :
100 S-UM 0,1,2,3,4,5,6,7:
110 :REM ALLE SPRITES EIN
120 DES BEGIN 128:REM 1.SPRITEDEF
130 :
140 SD "..........................."
150 SD "..........................."
160 SD "..........**************..."
170 SD "..................*........"
180 SD "..................*........"
190 SD "*.............**********.."
200 SD "**.........******...***.."
210 SD "******************...****"
220 SD "************************"
230 SD "............**********.."
240 SD "............*********..."
250 SD "................*....*..*"
260 SD "..........**********..."
270 SD "..........................."
280 :
290 DES END:REM KEINE 21 ZEILEN
300 :
310 DES BEGIN 129
320 :
330 SD "..........................."
340 SD "..........................."
350 SD "............**********.."
360 SD "..................*........"
370 SD "..................*........"
380 SD "*.............**********.."
390 SD "**.........******...***.."
400 SD "******************...****"
410 SD "************************"
420 SD "............**********..."
430 SD "............*********..."
440 SD "................*....*..*"
450 SD "..........**********..."
460 SD "..........................."
470 :
480 DES END
490 :
500 DES BEGIN 130
510 :
520 SD "..........................."
530 SD "..........................."
540 SD "............**********...."
550 SD "..................*........"
560 SD "..................*........"
570 SD "*.............**********.."
580 SD "**.........******...***.."
590 SD "******************...****"
600 SD "************************"
610 SD "............**********..."
620 SD "............*********..."
630 SD "................*....*..*"
640 SD "..........**********..."
650 SD "..........................."
660 :
670 DES END
680 :
690 DES BEGIN 131:REM 4.SPRITEDEF
700 :
720 SD "..........................."
730 SD "..........................."
740 SD "..,.............*****....."
750 SD "..................*........"
760 SD "..................*........"
770 SD "*.............**********.."
780 SD "**.........******...***.."
790 SD "******************...****"
800 SD "************************"
810 SD "............**********..."
820 SD "............*********..."
830 SD "................*....*..*"
840 SD "..........**********..."
850 SD "..........................."
860 :
870 DES END
880 :
890 DES BEGIN 132:REM 5.SPRITEDEF
900 :
920 SD "..........................."
930 SD "..........................."
940 SD "............**********..."
950 SD "..................*........"
960 SD "..................*........"
970 SD "*.............**********.."
980 SD "**.........******...***.."
990 SD "******************...****"
1000 SD "************************"
1010 SD "............**********..."
1020 SD "............*********..."
1030 SD "................*....*..*"
1040 SD "..........**********..."
1050 SD "..........................."
1060 :
1070 DES END
1080 :
1090 DES BEGIN 133:REM 6.SPRITEDEF
1100 :
1120 SD "..........................."
1130 SD "..........................."
1140 SD "..........**********..."
1150 SD "..................*........"
1160 SD "..................*........"
1170 SD "*.............**********.."
1180 SD "**.........******...***.."
1190 SD "******************...****"
1200 SD "************************"
1210 SD "............**********..."
1220 SD "............*********..."
1230 SD "................*....*..*"
1240 SD "..........**********..."
1250 SD "..........................."
1260 :
1270 DES END
1280 :
1290 DES BEGIN 134
1300 :
```

```
1320 SD "..........................."
1330 SD "..........................."
1340 SD "..........**************."
1350 SD "..................*........"
1360 SD "..................*........"
1370 SD "*............**********..."
1380 SD "**..........******...***.."
1390 SD "****************...*****"
1400 SD "**************************"
1410 SD "............**********..."
1420 SD "............*********...."
1430 SD "................*....*..*"
1440 SD "..........**************."
1450 SD "..........................."
1460 :
1470 DES END
1480 :
1490 FOR Z=0 TO 7
1500 X=INT(RND(1)*511):REM X-STARTPOSITION
1510 Y=INT(RND(1)*255):REM Y-STARTPOSITION
1520 RX=0:REM X-RICHTUNG
1530 RY=INT(RND(1)+.5):REM Y-RICHTUNG
1540 SX=INT(RND(1)*10)+3:REM ZEILENGESCHWINDIGKEIT
1550 SY=INT(RND(1)*10)+3:REM SPALTENGESCHWINDIGKEIT
1560 AS=INT(RND(1)*20)+5:REM ANIMATIONSGESCHWINDIGKEIT
1561 :
1562 SSET Z,128,5,RY:REM BLOCK=128;FARBE=5;PRIORITAET=0 ODER 1
1564 S-EXPAND Z,INT(RND(1)*3):REM VERGROESSERN
1565 SANIMATE Z,X,Y,RX,RY,SX,SY,128,5,AS
1566 NEXT Z
1567 SET TIMER 4000:REM 1KO INITIALISIEREN
1568 :
1570 FOR P=1 TO 200:NEXT
1580 FOR X=0 TO 39
1590 DISPLAY  X, 0,"*";
1600 DISPLAY  X,23,"*";
1610 NEXT
1620 :
1630 FOR Y=1 TO 23
1640 DISPLAY  0, Y,"*";
1650 DISPLAY 39, Y,"*";
1660 NEXT Y
1670 :
1680 DISPLAY 14,11,"ANIMATIONS-DEMO"
1690 :
1700 FOR Y=1 TO 23
1710 DISPLAY  0, Y," ";:GOSUB1860:DISPLAY  0, Y,"*";:GOSUB 1860
1720 NEXT Y
1730 :
1740 FOR X=0 TO 39
1750 DISPLAY  X,23," ";:GOSUB1860:DISPLAY  X,23,"*";:GOSUB1860
1760 NEXT X
1770 :
1780 FOR Y=23 TO 1 STEP-1
1790 DISPLAY 39, Y," ";:GOSUB1860:DISPLAY 39, Y,"*";:GOSUB1860
1800 NEXT Y
1810 :
1820 FOR X=39 TO 0 STEP-1
1830 DISPLAY  X, 0," ";:GOSUB1860:DISPLAY  X, 0,"*";:GOSUB1860
1840 NEXT X
1850 GOTO1700
1860 :
1870 FOR P=1 TO 3:NEXT
1880 RETURN
```

ASMON–Monitor C 64

ASMON ist ein sehr komfortabler Maschinensprachmonitor für den Commodore 64

Er belegt 4 kByte Speicherplatz an jeder beliebigen Stelle im Speicher durch die Verschiebbarkeit des Monitors.
Dieser Monitor hat im einzelnen folgende Befehle:

A xxxx <Mnemo> = Assembler.
Assembliert den eingegebenen mnemonischen Code an die Adresse xxxx und druckt die nächste Adresse aus. Wird kein Code eingegeben, so disassembliert der Monitor was in xxxx steht und druckt die nächste Adresse aus.

B [xxxx] = Einzelschrittabarbeitung.
Arbeitet das Maschinenprogramm ab xxxx auf Tastendruck in einzelnen Schritten entsprechender Befehle ab und zeigt dazu die Registerinhalte und den disassemblierten Befehl. Wird keine Adresse angegeben, so gilt die als PC gespeicherte Adresse. RUN/STOP = Abbruch.

D xxxx = Disassembler.
Disassembliert ab der Adresse xxxx auf Tastendruck. RUN/STOP = Abbruch. Fährt man mit dem Cursor in ein disassembliertes Programm hinein, so lassen sich Bytes ändern und nach Drücken von RETURN wird weiter disassembliert.

E xxxx, yyyy, zzzz = Adressenanpassung.
Paßt ein von xxxx nach zzzz verschobenes Programm an das Ursprungsprogramm an. Beispiel: Verschieben des Monitors von C000/D000 nach 9000/1000 und anpassen der Adressen!
T C000, D000, 9000
E C000, D000, 900

F xxxx, yyyy, zz = Fill-Befehl.
Füllt den Bereich von xxxx bis yyyy mit dem Byte zz.

G [xxxx] = GO-Befehl.
Springt zur Adresse xxxx. Wird keine Adresse angegeben, so wird zur im PC-angegebenen Adresse gesprungen.

H xxxx, yyyy, zz, * = Huntroutine.
Sucht Bytefolgen in dem Bereich von xxxx bis yyyy und gibt die Adressen, in denen sie stehen, aus. Als gesuchte Bytes können 8 mit Komma getrennte Bytes eingegeben werden, wobei als Joker für beliebige Bytes "*" benutzt werden können. Beispiel: H 9000, A000, 20, *, FF sucht zwischen 9000 und

A000 nach den Bytes 20, beliebiges Byte und FF.

L = LOAD lädt ein Programm von Kassette.
L "name" = lädt das Programm: "name" von Kassette.
L "name", xx = lädt das Programm: "name" aus Gerät xx.
L "name", xx, yyyy = lädt das Programm: "name" aus Gerät xx ab der Adresse yyyy.

MA xxxx = zeigt den Speicherinhalt und die entsprechenden **ASC-II-Zeichen** ab Adresse xxxx auf Tastendruck an. Die Bytes können verändert werden, indem man mit dem Cursor in das Memory-Dump hineinfährt, ändert, und RETURN drückt. RUN/STOP = Abbruch.

MB xxxx = Entsprechend MA nur mit Anzeige der Zeichen im PEEK/-POKE Code.

R = zeigt den derzeitigen Inhalt der 6510-Register an CPG/AG/XR/YR/P/SP und IRQ). Die Inhalte können sämtlich verändert werden.

S "name", xx, yyyy, zzzz = SAVE.
Speichert den Bereich von yyyy bis zzzz als Programm. "name" auf Gerät Nr. xx ab.

T xxxx, yyyy, zzzz = Transfer.
Verschiebt den Inhalt des Speicherbereiches von xxxx bis yyyy nach zzzz.

V ["name"[, xx [, yyyy]]] = Verify.
Entspricht dem LOAD-Format.

X = Exit.
Zurück ins BASIC.

Alle Fehlermeldungen werden durch ein **"?"** ausgedrückt.
Beim Anspringen des Monitors richtet er den BRK-Vektor auf sich selbst, d. h. erfolgt im Maschinenprogramm ein BRK, so wird wieder der Monitor angesprungen.

```
0 DATA8,141,237,205,142,238,205,140,239,205,104,141,241,205,76,41,192,104
1 DATA141,239,205,104,141,238,205,104,141,237,205,104,141,241,205,104,141
2 DATA244,205,104,141,245,205,186,142,240,205,173,244,205,208,3,206,245
3 DATA205,206,244,205,173,149,205,141,22,3,173,150,205,141,23,3,32,122
4 DATA192,32,103,205,32,129,205,176,248,160,0,162,0,221,203,205,240,14
5 DATA224,18,208,3,76,7,205,232,200,200,200,76,83,192,200,185,235,192,141
6 DATA117,192,200,185,235,192,141,118,192,32,120,193,76,71,192,160,255
7 DATA200,185,158,205,32,210,255,192,44,208,245,173,245,205,32,20,205,173
8 DATA244,205,32,20,205,230,211,230,211,173,237,205,32,20,205,230,211,173
9 DATA238,205,32,20,205,230,211,173,239,205,32,20,205,230,211,230,211,162
10 DATA7,173,241,205,141,243,205,14,243,205,144,13,169,49,32,210,255,224
11 DATA0,240,9,202,76,185,192,169,48,76,192,192,230,211,230,211,173,240
12 DATA205,32,20,205,230,211,230,211,173,21,3,32,20,205,173,20,3,32,20,205
13 DATA96,44,122,192,44,169,193,44,120,193,44,33,193,44,39,193,44,246,193
14 DATA44,188,194,44,191,194,44,229,194,44,116,195,44,14,198,44,130,200
15 DATA44,202,200,44,35,201,44,199,200,44,124,201,44,63,203,44,189,203,162
16 DATA248,154,76,134,227,32,188,204,141,245,205,32,188,204,141,244,205
17 DATA32,188,204,141,237,205,32,188,204,141,238,205,32,188,204,141,239
18 DATA205,160,7,32,129,205,201,49,240,3,24,144,1,56,46,241,205,192,0,240
19 DATA4,136,76,71,193,32,188,204,141,240,205,32,188,204,141,242,205,32
20 DATA188,204,120,141,20,3,173,242,205,141,21,3,88,96,32,129,205,176,33
21 DATA32,191,204,141,157,193,32,188,204,141,156,193,174,240,205,154,172
22 DATA239,205,173,241,205,72,173,237,205,174,238,205,40,76,0,128,173,245
23 DATA205,141,157,193,173,244,205,176,221,32,118,205,32,129,205,32,188
24 DATA204,141,252,205,32,188,204,141,253,205,32,129,205,32,188,204,141
25 DATA251,205,165,252,205,252,205,144,12,208,7,165,251,205,253,205,144
26 DATA3,76,7,205,160,0,173,251,205,145,251,165,251,205,253,205,208,8,165
27 DATA252,205,252,205,208,1,96,230,251,208,230,230,252,76,215,193,32,129
28 DATA205,201,66,240,3,169,0,44,169,1,141,254,205,32,118,205,32,103,205
29 DATA32,17,194,76,133,194,230,211,173,254,205,240,3,169,58,44,169,59,32
30 DATA210,255,32,108,205,169,0,141,255,205,230,211,172,255,205,177,251
31 DATA32,20,205,173,255,205,201,7,240,6,238,255,205,76,40,194,230,211,169
32 DATA0,141,255,205,172,255,205,177,251,172,254,205,208,68,201,32,144,8
33 DATA201,160,176,6,201,0,16,2,169,46,32,210,255,169,0,133,212,169,146
34 DATA32,210,255,173,255,205,201,7,240,6,238,255,205,76,70,194,165,251
35 DATA24,105,8,133,251,165,252,105,0,133,252,96,32,228,255,201,3,240,7
36 DATA201,0,240,245,76,8,194,96,201,0,48,22,141,0,206,41,63,14,0,206,44
37 DATA0,206,16,2,9,128,112,2,9,64,76,80,194,141,1,206,169,18,32,210,255
38 DATA173,1,206,76,152,194,169,0,44,169,1,141,254,205,32,118,205,169,0
39 DATA141,2,206,32,188,204,172,2,206,145,251,192,7,240,6,238,2,206,76,204
40 DATA194,169,0,133,211,76,17,194,32,188,204,133,250,32,188,204,133,249
41 DATA32,129,205,32,188,204,133,254,32,188,204,133,253,32,129,205,32,118
42 DATA205,160,0,185,250,197,252,144,40,208,6,165,249,197,251,144,32,177
43 DATA249,145,251,165,249,197,253,208,7,165,250,197,254,208,1,96,230,249
```

```
44 DATA208,2,230,250,230,251,208,229,230,252,76,18,195,165,253,56,229,249
45 DATA141,2,206,165,254,229,250,141,3,206,165,251,24,109,2,206,133,251
46 DATA165,252,109,3,206,133,252,177,253,145,251,165,253,197,249,208,7,165
47 DATA254,197,250,208,1,96,165,253,208,2,198,254,198,253,165,251,208,2
48 DATA198,252,198,251,76,80,195,32,118,205,160,0,140,35,206,32,129,205
49 DATA144,10,32,133,198,169,7,133,211,76,251,197,153,4,206,200,32,129,205
50 DATA176,96,153,4,206,200,32,129,205,176,87,153,4,206,173,155,205,133
51 DATA249,173,156,205,133,250,32,129,205,176,104,201,40,208,3,76,209,196
52 DATA201,65,208,19,32,129,205,144,3,76,16,196,160,160,140,249,205,32,227
53 DATA204,76,215,195,201,35,208,3,76,47,197,32,191,204,141,252,205,32,129
54 DATA205,201,44,240,74,201,13,240,57,32,191,204,141,253,205,32,129,205
55 DATA176,7,201,44,240,85,76,7,205,173,4,206,201,66,208,11,173,5,206,201
56 DATA82,240,4,201,73,208,88,169,0,162,3,76,62,197,169,10,162,1,76,62,197
57 DATA169,9,162,1,76,62,197,173,252,205,141,253,205,169,1,162,2,76,62,197
58 DATA162,2,173,252,205,141,253,205,32,129,205,201,88,240,7,201,89,240
59 DATA6,76,7,205,169,2,44,169,3,76,62,197,162,3,32,129,205,201,89,240,10
60 DATA201,88,240,3,76,7,205,169,4,44,169,5,76,62,197,165,251,24,105,2,141
61 DATA247,205,165,252,105,0,141,248,205,173,253,205,141,8,206,56,237,247
62 DATA205,141,253,205,173,252,205,141,9,206,237,248,205,141,252,205,162
63 DATA2,173,253,205,48,18,24,109,247,205,141,247,205,173,248,205,105,0
64 DATA141,248,205,76,189,196,169,0,56,237,253,205,141,2,206,173,247,205
65 DATA56,237,2,206,141,247,205,173,248,205,233,0,141,248,205,205,9,206
66 DATA208,12,173,247,205,205,8,206,208,4,169,8,208,112,76,7,205,32,129
67 DATA205,176,248,32,191,204,141,252,205,32,129,205,201,44,240,24,201,41
68 DATA240,46,32,191,204,141,253,205,32,129,205,201,41,208,71,162,3,169
69 DATA11,76,62,197,173,252,205,141,253,205,32,129,205,201,88,208,51,32
70 DATA129,205,201,41,208,44,162,2,169,6,208,41,173,252,205,141,253,205
71 DATA32,129,205,201,44,208,25,32,129,205,201,89,208,18,162,2,169,7,208
72 DATA15,32,188,204,141,253,205,162,2,169,12,208,3,76,7,205,141,7,206,142
73 DATA2,206,162,0,160,0,140,242,205,169,0,141,254,205,141,25,206,172,254
74 DATA205,185,4,206,172,242,205,217,39,206,208,15,238,242,205,173,254,205
75 DATA238,254,205,201,2,240,8,208,227,238,25,206,76,97,197,173,25,206,240
76 DATA8,224,55,240,188,232,76,75,197,142,3,206,162,0,160,0,236,3,206,240
77 DATA27,200,177,249,41,16,208,16,165,249,24,105,2,133,249,165,250,105
78 DATA0,133,250,76,136,197,232,76,150,197,200,177,249,41,15,205,7,206,240
79 DATA10,177,249,41,16,208,129,200,76,170,197,136,177,249,160,0,145,251
80 DATA230,251,208,2,230,252,174,2,206,224,1,240,26,173,253,205,145,251
81 DATA230,251,208,2,230,252,224,2,240,11,173,252,205,145,251,230,251,208
82 DATA2,230,252,32,103,205,169,65,32,210,255,230,211,32,108,205,230,211
83 DATA169,17,32,210,255,169,145,141,119,2,169,1,133,198,104,104,76,74,192
84 DATA32,118,205,169,0,141,35,206,32,35,198,32,228,255,240,251,201,3,208
85 DATA244,96,32,103,205,169,32,32,210,255,165,211,201,39,208,245,169,0
86 DATA133,211,230,211,169,46,32,210,255,230,211,32,108,205,169,20,133,211
87 DATA32,133,198,169,9,133,211,160,0,177,251,141,32,206,32,20,205,230,251
88 DATA208,2,230,252,224,1,240,36,230,211,177,251,141,33,206,32,20,205,230
89 DATA251,208,2,230,252,224,2,240,16,230,211,177,251,141,34,206,32,20,205
90 DATA230,251,208,2,230,252,96,173,155,205,133,249,173,156,205,133,250
91 DATA162,0,160,0,177,249,209,251,240,31,200,177,249,41,16,208,16,165,249
92 DATA24,105,2,133,249,165,250,105,0,133,250,76,145,198,224,55,240,29,232
93 DATA76,160,198,200,177,249,41,15,72,142,24,206,162,0,160,0,236,24,206
94 DATA240,34,200,200,200,232,76,197,198,173,35,206,208,13,169,63,32,210
95 DATA255,32,210,255,162,1,76,210,255,32,103,205,32,108,205,76,7,205,173
96 DATA35,206,208,22,185,39,206,32,210,255,200,185,39,206,32,210,255,200
97 DATA185,39,206,32,210,255,230,211,162,1,104,201,0,240,111,201,1,208,3
98 DATA76,151,199,201,2,208,3,76,170,199,201,3,208,3,76,188,199,201,4,208
99 DATA3,76,206,199,201,5,208,3,76,224,199,201,6,208,3,76,242,199,201,7
100 DATA208,3,76,14,200,201,8,208,3,76,42,200,201,9,208,1,96,201,10,208,11
101 DATA173,35,206,208,5,169,65,32,210,255,96,201,11,240,13,173,35,206,208
102 DATA69,169,35,32,210,255,76,151,199,173,35,206,208,36,169,40,32,210,255
103 DATA32,125,199,169,41,76,210,255,173,35,206,208,18,169,36,32,210,255
104 DATA160,2,177,251,32,20,205,136,177,251,32,20,205,232,232,96,173,35,206
```

```
105 DATA208,12,169,36,32,210,255,160,1,177,251,32,20,205,232,96,173,35,206
106 DATA208,249,32,151,199,169,44,32,210,255,169,88,76,210,255,173,35,206
107 DATA208,231,32,151,199,169,44,32,210,255,169,89,76,210,255,173,35,206
108 DATA208,193,32,125,199,169,44,32,210,255,169,88,76,210,255,173,35,206
109 DATA208,175,32,125,199,169,44,32,210,255,169,89,76,210,255,173,35,206
110 DATA208,177,169,40,32,210,255,32,151,199,169,44,32,210,255,169,88,32
111 DATA210,255,169,41,76,210,255,173,35,206,208,149,169,40,32,210,255,32
112 DATA151,199,169,41,32,210,255,169,44,32,210,255,169,89,76,210,255,173
113 DATA35,206,240,3,76,168,199,160,1,177,251,48,16,24,101,251,141,253,205
114 DATA165,252,105,0,141,252,205,76,96,200,169,0,56,241,251,141,253,205
115 DATA165,251,56,237,253,205,141,253,205,165,252,233,0,141,252,205,238
116 DATA253,205,208,3,238,252,205,238,253,205,208,3,238,252,205,169,36,32
117 DATA210,255,173,252,205,32,20,205,173,253,205,232,76,20,205,32,118,205
118 DATA32,188,204,160,0,145,251,200,32,129,205,141,25,206,169,18,197,211
119 DATA144,29,173,25,206,32,191,204,145,251,200,32,129,205,141,25,206,169
120 DATA18,197,211,144,8,173,25,206,32,191,204,145,251,169,0,133,211,32,38
121 DATA198,32,35,198,169,9,133,211,76,251,197,169,1,44,169,0,141,25,206
122 DATA32,12,201,32,129,205,176,34,32,87,201,32,129,205,201,44,208,24,32
123 DATA188,204,133,186,32,129,205,201,44,208,12,32,188,204,168,32,188,204
124 DATA170,169,0,133,185,32,103,205,173,25,206,32,213,255,176,28,32,183
125 DATA255,41,16,208,21,96,162,0,134,183,232,134,185,134,186,173,152,205
126 DATA133,187,173,153,205,133,188,96,76,7,205,32,12,201,32,129,205,32,87
127 DATA201,32,129,205,32,188,204,133,186,32,129,205,32,188,204,133,194,32
128 DATA188,204,133,193,32,129,205,32,188,204,133,175,32,188,204,133,174
129 DATA32,103,205,32,237,245,176,202,96,160,1,32,207,255,201,34,240,11,230
130 DATA183,136,153,8,206,200,200,76,89,201,192,17,176,177,96,173,245,205
131 DATA141,31,206,173,244,205,76,138,201,32,129,205,176,239,32,191,204,141
132 DATA31,206,32,188,204,141,30,206,169,0,141,35,206,174,240,205,154,32
133 DATA103,205,173,30,206,141,244,205,133,251,173,31,206,141,245,205,133
134 DATA252,32,103,205,32,103,205,32,103,205,32,122,192,169,234,141,113,202
135 DATA141,114,202,141,115,202,141,116,202,32,103,205,32,35,198,32,103,205
136 DATA138,72,32,228,255,240,251,201,3,208,3,76,14,205,104,170,173,32,206
137 DATA208,3,76,205,202,201,32,208,3,76,192,202,201,76,208,3,76,141,202
138 DATA201,108,208,3,76,156,202,201,64,208,3,76,231,202,201,96,208,3,76
139 DATA216,202,201,16,208,3,76,238,202,201,48,208,3,76,238,202,201,80,208
140 DATA3,76,238,202,201,112,208,3,76,238,202,201,144,208,3,76,238,202,201
141 DATA176,208,3,76,238,202,201,208,208,3,76,238,202,201,240,208,3,76,238
142 DATA202,173,32,206,141,113,202,224,1,240,16,173,33,206,141,114,202,224
143 DATA2,240,6,173,34,206,141,115,202,165,251,141,30,206,165,252,141,31
144 DATA206,173,241,205,72,174,238,205,172,239,205,173,237,205,40,234,234
145 DATA234,234,8,141,237,205,142,238,205,140,239,205,104,141,241,205,186
146 DATA142,240,205,76,153,201,76,45,203,173,33,206,141,30,206,173,34,206
147 DATA141,31,206,76,153,201,173,33,206,141,177,202,141,184,202,160,0,173
148 DATA34,206,141,178,202,141,185,202,185,0,0,141,30,206,200,185,0,0,141
149 DATA31,206,76,153,201,165,252,72,165,251,72,186,142,240,205,76,141,202
150 DATA238,244,205,208,3,238,245,205,76,14,205,104,141,30,206,104,141,31
151 DATA206,186,142,240,205,76,153,201,104,141,241,205,76,216,202,173,33
152 DATA206,48,30,165,251,24,109,33,206,141,2,206,165,252,105,0,141,3,206
153 DATA169,23,141,114,202,173,32,206,141,113,202,76,89,202,169,0,56,237
154 DATA33,206,141,253,205,165,251,56,237,253,205,141,2,206,165,252,233,0
155 DATA141,3,206,76,3,203,8,72,173,2,206,141,30,206,173,3,206,141,31,206
156 DATA104,76,118,202,162,0,32,118,205,32,129,205,32,188,204,141,3,206,32
157 DATA188,204,141,2,206,32,129,205,32,129,205,201,42,240,9,32,191,204,157
158 DATA8,206,169,0,44,169,1,157,16,206,32,129,205,176,4,232,76,86,203,224
159 DATA8,176,66,142,254,205,160,0,185,16,206,208,7,177,251,217,8,206,208
160 DATA9,204,254,205,240,35,200,76,125,203,32,228,255,201,3,240,14,165,251
161 DATA205,2,206,208,8,165,252,205,3,206,208,1,96,230,251,208,207,230,252
162 DATA76,123,203,32,103,205,32,108,205,76,146,203,76,7,205,32,188,204,141
163 DATA3,206,32,188,204,141,2,206,32,129,205,32,188,204,141,252,205,32,188
164 DATA204,141,253,205,32,129,205,32,118,205,173,253,205,56,237,2,206,141
165 DATA30,206,173,252,205,237,3,206,141,31,206,173,30,206,24,101,251,141
```

```
166 DATA30,206,173,31,206,101,252,141,31,206,160,0,140,25,206,169,1,141,35
167 DATA206,173,3,206,197,252,144,32,208,7,173,2,206,197,251,144,23,238,25
168 DATA206,173,2,206,56,229,251,141,247,205,173,3,206,229,252,141,248,205
169 DATA76,68,204,165,251,56,237,2,206,141,247,205,165,252,237,3,206,141
170 DATA248,205,32,133,198,165,251,205,30,206,208,8,165,252,205,31,206,208
171 DATA1,96,224,3,240,15,138,24,101,251,133,251,165,252,105,0,133,252,76
172 DATA68,204,160,2,173,252,205,209,251,144,232,208,8,136,173,253,205,209
173 DATA251,144,222,160,2,177,251,205,3,206,144,213,208,8,136,177,251,205
174 DATA2,206,144,203,160,1,173,25,206,208,19,177,251,24,109,247,205,145
175 DATA251,200,177,251,109,248,205,145,251,76,90,204,177,251,56,237,247
176 DATA205,145,251,200,177,251,237,248,205,145,251,76,90,204,32,129,205
177 DATA142,28,206,140,29,206,160,0,201,13,240,60,217,221,205,240,8,192,15
178 DATA240,51,200,76,199,204,152,10,10,10,10,141,249,205,32,129,205,160
179 DATA0,201,13,240,30,217,221,205,240,8,192,15,240,21,200,76,229,204,152
180 DATA13,249,205,141,27,206,172,29,206,174,28,206,173,27,206,96,230,211
181 DATA169,63,32,210,255,162,248,154,76,71,192,141,27,206,140,29,206,142
182 DATA28,206,173,27,206,160,0,140,246,205,170,42,42,42,42,42,41,15,205
183 DATA246,205,240,6,238,246,205,76,45,205,172,246,205,185,221,205,32,210
184 DATA255,138,41,15,160,0,140,246,205,205,246,205,240,6,238,246,205,76
185 DATA73,205,172,246,205,185,221,205,32,210,255,172,29,206,174,28,206,173
186 DATA27,206,96,169,13,76,210,255,165,252,32,20,205,165,251,76,20,205,32
187 DATA188,204,133,252,32,188,204,133,251,96,32,207,255,201,32,240,249,201
188 DATA36,240,245,201,13,24,208,2,56,96,96,44,17,192,44,8,206,44,207,206
189 DATA255,13,65,83,77,79,78,32,80,67,32,32,32,65,67,32,88,82,32,89,82,32
190 DATA32,78,86,45,66,68,73,90,67,32,32,83,80,32,32,73,82,81,13,42,32,32
191 DATA32,32,82,70,71,88,42,77,59,58,84,65,68,46,76,83,86,66,72,69,48,49
192 DATA50,51,52,53,54,55,56,57,65,66,67,68,69,70,0,0,0,246,48,114,0,59,144
193 DATA1,0,64,0,0,0,127,255,0,7,0,0,0,64,83,84,89,4,82,65,84,77,79,78,32
194 DATA42,48,48,48,32,32,32,32,32,41,0,63,0,2,8,0,144,185,158,205,1,32,48
195 DATA56,66,67,67,66,67,83,66,69,81,66,78,69,66,77,73,66,80,76,66,86,67
196 DATA66,86,83,66,82,75,67,76,67,83,69,67,67,76,68,83,69,68,67,76,73,83
197 DATA69,73,67,76,86,73,78,88,73,78,89,68,69,88,68,69,89,73,78,67,68,69
198 DATA67,78,79,80,80,72,65,80,76,65,80,72,80,80,76,80,84,83,88,84,88,83
199 DATA84,65,89,84,89,65,84,65,88,84,88,65,76,68,65,76,68,89,76,68,88,83
200 DATA84,65,83,84,89,83,84,88,82,84,83,82,84,73,74,77,80,74,83,82,69,79
201 DATA82,79,82,65,65,78,68,66,73,84,76,83,82,65,83,76,82,79,82,82,79,76
202 DATA83,66,67,65,68,67,67,77,80,67,80,88,67,80,89,144,24,176,24,240,24
203 DATA208,24,48,24,16,24,80,24,112,24,0,25,24,25,56,25,216,25,248,25,88
204 DATA25,120,25,184,25,232,25,200,25,202,25,136,25,238,0,230,1,246,2,254
205 DATA20,206,0,198,1,214,2,222,20,234,25,72,25,104,25,8,25,40,25,186,25
206 DATA154,25,168,25,152,25,170,25,138,25,169,12,173,0,165,1,181,2,189,4
207 DATA185,5,161,6,177,23,160,12,172,0,164,1,180,2,188,20,162,12,174,0,166
208 DATA1,182,3,190,21,141,0,133,1,149,2,157,4,153,5,129,6,145,23,140,0,132
209 DATA1,148,18,142,0,134,1,150,3,158,21,96,25,64,25,76,0,108,27,32,16,73
210 DATA12,77,0,69,1,85,2,93,4,89,5,65,6,81,23,9,12,13,0,5,1,21,2,29,4,25
211 DATA5,1,6,17,23,41,12,45,0,37,1,53,2,61,4,57,5,33,6,49,23,44,0,36,17
212 DATA78,0,70,1,86,2,94,4,74,26,14,0,6,1,22,2,30,4,10,26,110,0,102,1,118
213 DATA2,126,4,106,26,46,0,38,1,54,2,62,4,42,26,233,12,237,0,229,1,245,2
214 DATA253,4,249,5,225,6,241,23,105,12,109,0,101,1,117,2,125,4,121,5,97
215 DATA6,113,23,201,12,205,0,197,1,213,2,221,4,217,5,193,6,209,23,224,12
216 DATA236,0,228,17,192,12,204,0,196,17,255,0,-1
220 :
230 :
240 :REM --- EINLESEN DES MONITORS ----
250 :
260 :I=49152
270 :READ A
280 :IF A=-1 THEN SYS49152
290 :POKEI,A
300 :I=I+1
310 :GOTO 270
```

```
------ DIE BEFEHLE ----------

MA XXXX     = MEMORY MIT ASCII

H XXXX,YYYY,ZZ,*,... = HUNT
  MIT JOKERS

B [XXXX]    = EINZELSCHRITT
  SIMULATOR

G [XXXX]    = GOTO

L ["NAME"[,XX[,YYYY]]] = LOAD

V ["NAME"[,XX[,YYYY]]] = VERIFY

S "NAME",XX,YYYY,ZZZZ = SAVE

R           = REGISTERINHALTE

X           = BASIC

MB XXXX     = MEMORY MIT BKODES

A XXXX ...  = ASSEMBLER

D XXXX      = DISASSEMBLER

F XXXX,YYYY,ZZ = FILL ROUTINE

T XXXX,YYYY,ZZZZ = TRANSFER

E XXXX,YYYY,ZZZZ = ADDRESSEN ANPASSEN
```

```
**********************************
****                          ****
****   A S M O N -- 6 4      ****
****                          ****
****   (C) 1984 ONLY BY       ****
****   STEPHAN BARTOSCH       ****
****    ALTRHEINSTR. 25       ****
****    6843 BIBLIS - 2       ****
****                          ****
****                          ****
**********************************
```

Diskloader C 64

Das hier vorgestellte Basic-Programm zeigt das Directory einer Diskette und erlaubt das Starten eines Programms per Knopfdruck. Dieses Programm sollte als erstes Programm auf alle Disketten abgelegt werden, da man so die Möglichkeit besitzt, es mit LOAD "+",8 zu laden. Nach einem RUN-Befehl erscheint auf dem Bildschirm das zweispaltige Directory, wobei bei jedem Programm ein Kennbuchstabe angezeigt wird. Durch Drücken eines beliebigen Kennbuchstabens wird das entsprechende Programm geladen und automatisch gestartet.
Das Programm hat gegenüber vielen anderen Menü-Programmen den Vorteil, daß es sich ohne Programmänderung an jede Disk anpaßt. Es können bis zu 36 Programme verwaltet werden.

```
10 RUN 100
100 REM DIM FUER MAXIMAL 40 FILES
110 :
120 DIM N$(40)
130 :
500 POKE 53281,1:POKE 53280,2
510 PRINT"■■■■■"TAB(13)"■BITTE WARTEN"
520 PRINT"■■"TAB(9)"DIRECTORY WIRD GELESEN"
900 :
910 REM DIRECTORY EINLESEN
920 :
1005 Q=0
1010 OPEN 15,8,15,"IO":OPEN 2,8,2,"#"
1020 T=18:S=1
1030 PRINT#15,"U1";2;0;T;S
1040 PRINT#15,"B-P";2;0
1050 GET#2,X$:IF X$="" THEN X$=CHR$(0)
1060 T=ASC(X$)
1070 GET#2,X$:IF X$="" THEN X$=CHR$(0)
1080 S=ASC (X$)
1090 FORX=0 TO 7
1100 PRINT#15,"B-P";2;X*32+2
1110 GET #2,X$:IF X$="" THEN X$=CHR$(0)
1120 IF ASC (X$)=0 THEN 1220
1130 PRINT#15,"B-P";2;X*32+5
1140 FF$=""
1150 FORY = 0 TO 15
1160 GET#2,X$:IF X$="" THEN X$=CHR$(0)
1170 IF ASC(X$)=160 THEN 1212
1180 FF$=FF$+X$
1190 NEXTY
1212 N$(Q)=FF$:Q=Q+1
```

```
1220 NEXTX
1230 IF T<>0 THEN 1030
1240 CLOSE1:CLOSE15
2000 :
2010 PRINT"■■ C64-DISKLOADER                          ■"
2020 T=1
2030 FORI= 0 TO Q-1
2035 IF I=18 THEN PRINT"■■":T=21
2040 PRINTTAB(T);CHR$(65+I);" "N$(I)
2100 NEXTI
2110 PRINT"■":FORI=1 TO 18:PRINT:NEXTI
2115 PRINT"■────────────────────────────────────────";
2120 PRINT"■ES BEFINDEN SICH "Q"FILES AUF DER DISK"
2130 PRINT"■  ■WELCHES FILE SOLL GESTARTET WERDEN ?";
2200 GET E$:IF E$="" THEN 2200
2210 I=ASC(E$)-65
2215 IF I<0 THEN 2200
2220 IF I>(Q-1)THEN 2200
2230 PRINT"■■LOAD "CHR$(34)N$(I)CHR$(34)",8"
2240 PRINT"■■■■POKE 53281,6:RUN"
2250 POKE 631,19:POKE 632,13:POKE 633,13:POKE 198,3:END
```

C 64 Directory anzeigen ohne Programmverlust

Was bei den meisten Computern Standard ist, das Directory anzeigen ohne das im Speicher abgelegte Programm zu zerstören, kann auch dem C 64 beigebracht werden.

In dem Speicherbereich des Commodore 64 gibt es einen Zeiger, welcher auf die Befehlsdecodierung zeigt. Unser Programm nutzt dies aus und läßt den Zeiger (0308/0309) umbiegen, dadurch ist es möglich, einen weiteren Basic-Befehl zu implementieren.

Unser neuer Befehl (!) kann wie jeder andere Basic-Befehl in Programmen oder auch direkt eingegeben werden.

Nach dem Abtippen des Programms wird dieses mit SYS 49152 initialisiert. Nach dem Erscheinen der READY-Meldung ist der neue Befehl aktiviert. Trifft der Basic-Interpreter auf ein Anführungszeichen (!), so wird automatisch das Directory angezeigt, ohne daß das Basic-Programm unterbrochen bzw. zerstört wird.

Für den etwas erfahreneren Programmierer ist das Programm als gut dokumentiertes Assembler-Listing abgedruckt.

Das Programm belegt den Speicherbereich von 49152 bis 49285, also kein Basic-Bereich.

```
1 REM ***********************************
2 REM *                                 *
3 REM * DIRECTORY OHNE PROGRAMMVERLUST*
4 REM *                                 *
5 REM *   COPYRIGHT BY R.PETER          *
6 REM *                                 *
7 REM *   (C) 03.01.85                  *
8 REM *                                 *
9 REM ***********************************
60000 DIMH(75):FORI=0TO9
60010 H(48+I)=I:H(65+I)=I+10:NEXT
60020 FORI= 49152 TO 49286 :READA$
60030 H=ASC(LEFT$(A$,1))
60040 L=ASC(RIGHT$(A$,1))
60050 D=H(H)*16+H(L):S=S+D:POKEI,D
60060 A=A+1:IFA<20THENNEXT:A=-1
60070 READV:Z=Z+1:IFV=STHEN60085
60080 PRINT"DATAFEHLER IN ZEILE :";60200+Z:END
```

```
60085 IFA<0THEN60100
60090 S=0:A=0:NEXT
60100 END
60201 DATA A9,0B,8D,08,03,A9,C0,8D,09,03,60,20,73,00,C9,21,F0,06,20,79,  1722
60202 DATA 00,4C,E7,A7,A9,24,85,FB,A9,FB,85,BB,A9,00,85,BC,A9,01,85,B7,  2779
60203 DATA A9,08,85,BA,A9,60,85,B9,20,D5,F3,A5,BA,20,B4,FF,A5,B9,20,96,  2917
60204 DATA FF,A9,00,85,90,A0,03,84,FB,20,A5,FF,85,FC,A4,90,D0,2F,20,A5,  2844
60205 DATA FF,A4,90,D0,28,A4,FB,88,D0,E9,A6,FC,20,CD,BD,A9,20,20,D2,FF,  3345
60206 DATA 20,A5,FF,A6,90,D0,12,AA,F0,06,20,D2,FF,4C,64,C0,A9,0D,20,D2,  2693
60207 DATA FF,A0,02,D0,C6,20,42,F6,20,73,00,4C,E7,A7,00,  1788
61000 REM
61010 REM START MIT SYS 49152
```

Maschinenprogramm

```
2
                        ;
16:     C000                    .OPT P1,OO
                        ;
                        ; ******************************
                        ; *                            *
                        ; *   C64 -  C A T A L O G      *
                        ; *                            *
                        ; * DIRECTORY OHNE PROGRAMM-    *
                        ; * VERLUST EINLESEN.          *
                        ; * DIESE ROUTINE KANN MAN     *
                        ; * ALS UNTERROUTINE IN BASIC  *
                        ; * PROGRAMMEN VERWENDEN !     *
                        ; *                            *
                        ; * COPYRIGHT 85 BEI R.PETER   *
                        ; ******************************
                        ;
110:    C000                    *=    49152   ;PROGRAMMSTART = $C000
                        ;
                        ; SYSTEM ADRESSEN
130:    C000
                        ;
150:    C000            CHRGET  =     $73     ;CHRGET-ROUTINE HOLT ASCII ZEICHEN
160:    C000            CHRGOT  =     $79     ;CHRGET-ROUTINE FLAG WIEDER SETZEN
170:    C000            INTERP  =     $A7E7   ;IN DEN BASIC-INTERPRETER
                         ;
190:    C000            FNADR   =     $BB     ;ADRESSE DES FILENAMENS
200:    C000            FNLEN   =     $B7     ;LAENGE DES FILENAMENS
210:    C000            FA      =     $BA     ;GERAETENUMMER
220:    C000            SA      =     $B9     ;SEKUNDAERADRESSE
230:    C000            SENDNAM =     $F3D5   ;FILE MIT NAMEN ERDEFFNEN
240:    C000            TALK    =     $FFB4   ;TALK SENDEN
250:    C000            SECTALK =     $FF96   ;SEKUNDAERADRESSE SENDEN
260:    C000            STATUS  =     $90     ;FLOPPY STATUS
270:    C000            IECIN   =     $FFA5   ;BYTE VOM IEC-BUS HOLEN
280:    C000            LNPRT   =     $BDCD   ;PRINT 16 BIT ZAHL
290:    C000            PRINT   =     $FFD2   ;PRINT AKKU
300:    C000            CLSFIL  =     $F642   ;CLOSE
                         ;
                         ;
                         ; DEN BEFEHL "!" IN DAS BASIC
                         ; EINBINDEN UND INITIALISIEREN
                         ;
400:    C000 A9 0B      INIT    LDA   #<DECODE  ;POINTER AUF NEUE ABFRAGE
410:    C002 8D 08 03           STA   $0308
```

```
420:   C005 A9 C0              LDA  #>DECODE
430:   C007 8D 09 03           STA  $0309
440:   C00A 60                 RTS
                        ;
                        ; DECODIEREN DES BEFEHLES
                        ;
520:   C00B 20 73 00 DECODE    JSR  CHRGET   ;NAECHSTE ZEICHEN HOLEN
530:   C00E C9 21              CMP  #"!"     ;MIT SONDERZEICHEN VERGLEICHEN
540:   C010 F0 06              BEQ  FOUND    ;DIRECTORY ANZEIGEN
550:   C012 20 79 00           JSR  CHRGOT   ;FLAGS WIEDER SETZEN
560:   C015 4C E7 A7           JMP  INTERP   ;ZURUECK IN BASIC-INTERPRETER
                        ;
                        ; DIRECTORY ANZEIGEN
                        ;
620:   C018 A9 24     FOUND    LDA  #"$"     ;DOLLARZEICHEN ALS FILENAME
630:   C01A 85 FB              STA  $FB      ;ABLEGEN
640:   C01C A9 FB              LDA  #$FB     ;ADRESSE DES FILENAMENS
650:   C01E 85 BB              STA  FNADR
660:   C020 A9 00              LDA  #0       ;HOEHERWERTIGES BYTE
670:   C022 85 BC              STA  FNADR+1
680:   C024 A9 01              LDA  #1       ;LAENGE DES FILENAMENS
690:   C026 85 B7              STA  FNLEN
700:   C028 A9 08              LDA  #8       ;GERAETENUMMER DES LAUFWERKS
710:   C02A 85 BA              STA  FA
720:   C02C A9 60              LDA  #$60     ;SEKUNDAERADRESSE FUER LOAD
730:   C02E 85 B9              STA  SA
740:   C030 20 D5 F3           JSR  SENDNAM  ;FILE MIT NAMEN ERÖFFNEN
750:   C033 A5 BA              LDA  FA
760:   C035 20 B4 FF           JSR  TALK     ;TALK SENDEN
770:   C038 A5 B9              LDA  SA
780:   C03A 20 96 FF           JSR  SECTALK  ;SENDE SEKUNDAERADRESSE
790:   C03D A9 00              LDA  #0
800:   C03F 85 90              STA  STATUS   ;STATUS LOESCHEN
810:   C041 A0 03              LDY  #3       ;ERSTEN DREI BYTE UEBERLESEN
820:   C043 84 FB     LOOP1    STY  $FB      ;ALS ZAEHLER MERKEN
830:   C045 20 A5 FF           JSR  IECIN    ;BYTE VOM FLOPPY HOLEN
840:   C048 85 FC              STA  $FC      ;UND MERKEN
850:   C04A A4 90              LDY  STATUS   ;STATUS TESTEN
860:   C04C D0 2F              BNE  LOOP4
870:   C04E 20 A5 FF           JSR  IECIN    ;BYTE VOM FLOPPY HOLEN
880:   C051 A4 90              LDY  STATUS   ;STATUS TESTEN
890:   C053 D0 28              BNE  LOOP4
900:   C055 A4 FB              LDY  $FB      ;ZAEHLER HOLEN
910:   C057 88                 DEY           ;UND ERNIEDRIGEN
920:   C058 D0 E9              BNE  LOOP1
930:   C05A A6 FC              LDX  $FC      ;BYTE ZURUECKHOLEN
940:   C05C 20 CD BD           JSR  LNPRT    ;16 BIT ZAHL AUSGEBEN
950:   C05F A9 20              LDA  #" "     ;ZAHL DER BELEGTEN BLOCKS
960:   C061 20 D2 FF           JSR  PRINT    ;LEERZEICHEN AUSGEBEN
970:   C064 20 A5 FF LOOP3     JSR  IECIN    ;NAECHSTE BYTE HOLEN
980:   C067 A6 90              LDX  STATUS   ;STATUS TESTEN
990:   C069 D0 12              BNE  LOOP4
1000:  C06B AA                 TAX           ;BYTE TESTEN
1010:  C06C F0 06              BEQ  LOOP2    ;NULL PRINT DANN ZEILENENDE
1020:  C06E 20 D2 FF           JSR  PRINT    ;SONST AUSGEBEN
1030:  C071 4C 64 C0           JMP  LOOP3    ;UND NAECHSTES ZEICHEN HOLEN
1040:  C074 A9 0D     LOOP2    LDA  #13      ;ZEILENVORSCHUB
1050:  C076 20 D2 FF           JSR  PRINT    ;AUSGEBEN
1060:  C079 A0 02              LDY  #2       ;2 BYTES FUER LINKADRESSE
1070:  C07B D0 C6              BNE  LOOP1
1080:  C07D 20 42 F6 LOOP4     JSR  CLSFIL   ;CLOSE
1085:  C080 20 73 00           JSR  CHRGET
1090:  C083 4C E7 A7           JMP  INTERP   ;IN BASIC INTERPRETER
```

Data-Generator für C 64

Ein Data-Generator der einen erweiterten Funktionsbereich aufweist!

Dies ist die Entwicklung eines DATA-GENERATORS der u. a. auch Prüfsummen, Copyrightkopf, Löschen des Generators enthält.
Nach dem Starten des Programms meldet sich der Data-Generator mit der Frage: „Copyright (J/N)?“ Beim Betätigen der J-Taste folgen die Fragen nach dem Programmnamen, Namen des Autors und das Datum. Jede Eingabe muß mit RETURN abgeschlossen werden.
Beim Betätigen der N-Taste wird der Copyrightkopf gelöscht und man kann im Programm weiterfahren. Anschließend wird nach der Anfangs- und Endadresse des Maschinenprogramms gefragt.
Nach dieser Eingabe werden die Data-Zeilen erzeugt. Ist dieser Vorgang beendet, wird wieder nach der Anfangs- und Endadresse gefragt, in die das Programm von Basic aus eingelesen werden soll. Danach folgt die Frage, ob das Programm direkt nach dem EINPOKEN gestartet werden soll.
Bei Betätigen der N-Taste folgt das Löschen des Data-Generators. Danach listet sich das generierte Programm und kann auf Kassette oder Diskette gespeichert werden.

```
10 POKE53281,0:POKE53280,0
11 PRINT"▯▯▯▯▯▮▮▮*** DATA-PROGRAMM-GENERATOR ***"
12 PRINT"▯▯▯▯▯▯▯▯▯▯▮▮▮▮▮▮▮▮▮*** COPYRIGHT (J/N) ***"
14 GETA$:IFA$=""THEN14
15 IFA$="J"THEN21
16 IFA$="N"THEN19
17 GOTO14
18 POKE631,19:FORI=632TO640:POKEI,13:NEXT:POKE198,7:END
19 PRINT"▯";:FORI=1TO4:PRINTI:NEXT:PRINT"GOTO20":GOTO18
20 PRINT"▯";:FORI=5TO9:PRINTI:NEXT:PRINT"RUN30":GOTO18
21 PRINT"▯";:INPUT"PROGRAMMNAME:";B$
22 INPUT"NAME DES AUTOR:";C$
23 INPUT"DATUM:";D$
24 PRINT"▯3 REM *    ";B$;:PRINTTAB(38)"*"
25 PRINT"5 REM *    COPYRIGHT BY ";C$;:PRINTTAB(38)"*"
26 PRINT"7 REM *    (C) ";D$;:PRINTTAB(38)"*"
27 PRINT"RUN30"
28 POKE631,19:FORI=632TO635:POKEI,13:NEXT:POKE198,5:END
30 PRINT"▯";:INPUT"ANFANGSADRESSE:";A
31 INPUT"ENDADRESSE";E
32 H=INT(A/256):L=A-H*256:POKE640,H:POKE641,L
33 H=INT(E/256):L=E-H*256:POKE642,H:POKE643,L
34 POKE2,1:IFA>ETHEN30
35 A=PEEK(640)*256+PEEK(641)
36 E=PEEK(642)*256+PEEK(643)
37 DIMH$(20):FORI=0TO9:H$(I)=CHR$(I+48)
38 H$(I+10)=CHR$(I+65):NEXT:Z=3
39 FORI=1TO20:D=PEEK(A):S=S+D
40 H=INT(D/16):L=D-16*H
41 A$=A$+H$(H)+H$(L)+","
42 IFA=ETHENZ=2:D=PEEK(2):GOTO44
43 A=A+1:NEXT:D=PEEK(2):POKE2,D+1
44 A$=STR$(60200+D)+" DATA "+A$
45 A$=A$+STR$(S)
46 H=INT(A/256):L=A-H*256:POKE640,H:POKE641,L
47 IFZ=2THEN49
48 PRINT"▯";A$:PRINT"RUN35":GOTO50
49 PRINT"▯";A$:PRINT"RUN51":GOTO50
50 POKE631,19:POKE632,13:POKE633,13:POKE198,3:END
51 PRINT"▯";:INPUT"B-ANFANGSADRESSE:";A
52 INPUT"B-ENDADRESSE:";E:IFE<ATHEN51
53 PRINT"▯60020 FORI="A"TO"E":READA$":PRINT"GOTO54":GOTO50
```

```
54 PRINT"■■■■■■SOLL DAS PROGRAMM DIREKT NACH DEM LADEN"
55 PRINT"■■GESTARTET WERDEN ?"
56 GETA$:IFA$=""THEN56
57 IFA$="J"THEN60
58 IFA$="N"THEN62
59 GOTO56
60 PRINT"■";:INPUT"STARTADRESSE:";S
61 PRINT"■60100 SYS"S":END":PRINT"GOTO70":GOTO50
62 PRINT"■60100 END":PRINT"GOTO70":GOTO50
70 POKE2,10
71 IFPEEK(2)>62THEN75
72 PRINT"■";:FORI=1TO7:PRINTPEEK(2):POKE2,PEEK(2)+1
73 NEXTI:PRINT"GOTO71":GOTO74
74 POKE631,19:FORI=632TO640:POKEI,13:NEXTI:POKE198,9:END
75 PRINT"■";:FORI=70TO75:PRINTI:NEXTI:PRINT"LIST":GOTO74
60000 DIMH(75):FORI=0TO9
60010 H(48+I)=I:H(65+I)=I+10:NEXT
60030 H=ASC(LEFT$(A$,1))
60040 L=ASC(RIGHT$(A$,1))
60050 D=H(H)*16+H(L):S=S+D:POKEI,D
60060 A=A+1:IFA<20THENNEXT:A=-1
60070 READV:Z=Z+1:IFV=STHEN60085
60080 PRINT"DATAFEHLER IN ZEILE :";60200+Z:END
60085 IFA<0THEN60100
60090 S=0:A=0:NEXT
```

SAVING DATA-GENERATOR

```
1 REM ************************************
2 REM *                                  *
3 REM *   DEMO ZU DATA-GENERATOR         *
4 REM *                                  *
5 REM *   COPYRIGHT BY TRONIC-VERLAG     *
6 REM *                                  *
7 REM *   (C) 14.01.84                   *
8 REM *                                  *
9 REM ************************************
60000 DIMH(75):FORI=0TO9
60010 H(48+I)=I:H(65+I)=I+10:NEXT
60020 FORI= 832 TO 900 :READA$
60030 H=ASC(LEFT$(A$,1))
60040 L=ASC(RIGHT$(A$,1))
60050 D=H(H)*16+H(L):S=S+D:POKEI,D
60060 A=A+1:IFA<20THENNEXT:A=-1
60070 READV:Z=Z+1:IFV=STHEN60085
60080 PRINT"DATAFEHLER IN ZEILE :";60200+Z:END
60085 IFA<0THEN60100
60090 S=0:A=0:NEXT
60100 END
60201 DATA 10,44,41,54,41,2D,47,45,4E,45,52,41,54,4F,52,20,20,20,20,20,  1182
60202 DATA 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,  640
60203 DATA 20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,  640
60204 DATA 20,20,20,20,20,20,20,20,20,  288
```

Tape Directory für den C 64

Directory für den C 64 mit Datasette

Dieses Programm bietet allen C 64-Besitzern, die sich den Luxus einer Diskettenstation noch nicht leisten können, die Möglichkeit, Programme von der Datasette mit Hilfe eines Directory einladen zu können.
Nach der Eingabe des Programms, welches mit „RUN" gestartet wird, erscheint auf dem Bildschirm eine Kurzanleitung zum Programm selbst. Drücken Sie nun die „RETURN"-Taste, so druckt Ihnen der Computer die erste Seite des Directory aus, wobei die Umschaltung zu weiteren Seiten wiederum durch Drücken von „RETURN" erreicht wird. Haben Sie die von Ihnen gewünschte Seite auf dem Bildschirm, so erwartet der Computer nun die Eingabe der Zahl, die dem von Ihnen gesuchten Programm entspricht. Nach der Eingabe dieser Zahl erwartet der Computer von Ihnen das Drücken der Vorspultaste der Datasette (F.FWD). Die Datasette spult nun bis zum gewünschten Programm vor und stoppt automatisch. Der Computer fordert Sie nun zum Betätigen der Stop-Taste auf, woraufhin durch Drücken der Play-Taste das Programm geladen wird.
Tape Directory sollte am Anfang jeder Kassettenseite abgespeichert werden. Die Programmnamen müssen in der Reihenfolge, in der sie auf der Kassette gespeichert sind, in die Datazeilen am Ende des Programms eingegeben werden. Die Programmnamen dürfen jedoch nicht länger als 60 Zeichen sein. Befinden sich im Programm Kommas bzw. Doppelpunkte, so müssen diese in Anführungszeichen gesetzt werden. Hinter den Namen des Programms wird, durch ein Komma getrennt, eine sechsstellige Zahl eingegeben, welche die Vorspuldauer in sechzigstel Sekunden angibt.

Wichtig!
Die Programmnamen auf der Kassette müssen genau mit den Namen in den Data-Zeilen übereinstimmen!

Erklärung der Steuerzeichen:

■	=	CTRL + RVS ON
■	=	CTRL + RVS OFF
■	=	SHIFT + CLR/HOME
■	=	CLR/HOME
■	=	HELLBLAU
■	=	CURSOR DOWN
■	=	CURSOR LEFT

```
10 REM
20 REM              TAPE - DIRECTORY
30 REM              BY  C. OSTERKAMP
40 REM
50 REM
100 GOSUB 1000
110 GOSUB 1500
120 GOSUB 2000
200 PRINT"■"
210 PRINT"■";B1$;G1$;B2$:PRINT:PRINT
220 FOR I=J+1 TO 12+J
230 PRINT"    ■■■";I;TAB(4)AL$(I);TAB(19)I+12;TAB(22)AL$(I+12)
240 NEXTI
250 IF J=0 THEN PRINT:PRINT:PRINT:PRINTB1$;G2$;B2$;:GOTO 2500
260 PRINT:PRINT:PRINT:PRINTB1$;G3$;B2$;:GOTO 2500
270 GOSUB 3000
1000 REM   ***   DATEN EINLESEN   ***
1010 DIM A$(48),A(48),AL$(48),G$(2)
1020 I=I+1:READ A$(I),A(I):AL$(I)=LEFT$(A$(I),15)
1030 IF I=48 THEN RETURN
1040 GOTO 1020
1500 REM  **  VARIABLEN DEFINIEREN  **
1510 B1$="┌──────────────────────────────────────┐"
1520 B2$="└──────────────────────────────────────┘"
1530 C1$="|                          ■      ■     | "
1540 C2$="|  RECORD  PLAY  REWIND ■ F.FWD ■ STOP  | "
1550 C3$="■■■PRESS F.FWD ON TAPE"
1560 D1$="|                              ■      ■ | "
1570 D2$="|  RECORD  PLAY  REWIND  F.FWD ■ STOP ■ | "
1580 D3$="■■■PRESS STOP ON TAPE "
1590 E1$="|          ■      ■                    | "
1600 E2$="|  RECORD ■ PLAY ■ REWIND  F.FWD  STOP  | "
1610 F$="■OK"
```

```
1620 G1$="|  *****    TAPE - DIRECTORY   *****  | "
1630 G2$="|  *****        SEITE  1       *****  | "
1640 G3$="|  *****        SEITE  2       *****  | "
1700 RETURN
2000 REM   ***   ANLEITUNG    ***
2010 POKE 53280,0:POKE 53281,0:PRINT"▮▮";B1$;G1$;B2$
2020 PRINT"▮'TAPE - DIRECTORY' ERMOEGLICHT  IHNEN"
2030 PRINT"▮EINEN  SCHNELLEN UND  KOMFORTABLEN ZU-"
2040 PRINT"▮GRIFF  AUF  IHRE  PROGRAMME.
2050 PRINT"▮GEBEN  SIE  LEDIGLICH  DIE  PROGRAMM -"
2060 PRINT"▮ZIFFER  EIN ,  UND  DAS PROGRAMM  WIRD"
2070 PRINT"▮AUTOMATISCH  GELADEN."
2080 PRINT"▮<<RETURN>>  ERGIBT  DIE  NAECHSTE  DI-"
2090 PRINT"▮RECTORY - SEITE.
2100 PRINT"▮▮";B1$;"|  *****          <RETURN>          *****  | ";LEFT$(B2$,39);
2110 POKE 198,0:WAIT 198,1:RETURN
2500 REM   ***   EINGABE ABFRAGEN   ***
2510 POKE 198,0
2520 GET X$:IF X$="" THEN 2520
2530 IF ASC(X$)=13 THEN J=24-J:GOTO 210
2540 IF ASC(X$)<48 OR ASC(X$)>57 THEN 2520
2550 GET Y$:IF Y$="" THEN 2540
2560 IF ASC(Y$)=13 THEN 2610
2570 IF ASC(Y$)<48 OR ASC(Y$)>57 THEN 2550
2580 X=VAL(X$):Y=VAL(Y$)
2590 Z=(X*10)+Y:IF Z>48 THEN 2510
2600 GOTO 270
2610 Z=VAL(X$):GOTO 270
3000 REM   ***   PROGRAMM SUCHEN   ***
3010 IF (PEEK(1)AND16)<>0 THEN 3040
3020 PRINT"▮";B1$;D1$;D2$;D1$;B2$
3030 PRINTD3$:WAIT 1,24:PRINTF$:FOR I=1 TO 500:NEXT I
3040 PRINT"▮";B1$;C1$;C2$;C1$;B2$
3050 PRINTC3$
3060 IF (PEEK(1)AND 16)<>0 THEN 3060
3070 TI$="000000":PRINTF$
3080 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮";(A(Z)),TI:IF TI<A(Z) THEN 3080
3090 POKE 192,1:POKE 1,PEEK(1)OR32
3100 FOR I=1 TO 500:NEXT I
3110 PRINT"▮";B1$;D1$;D2$;D1$;B2$
3120 PRINTD3$:WAIT 1,24:PRINTF$:FOR I=1 TO 500:NEXT I
3130 PRINT"▮";B1$;E1$;E2$;E1$;B2$
3140 PRINT"▮▮▮▮▮▮▮▮▮LOAD";CHR$(34)+A$(Z)+CHR$(34)
3150 POKE 631,19:POKE 632,17:POKE 633,17:POKE 634,17:POKE 635,17:POKE 636,17
3160 POKE 637,17:POKE 638,17:POKE 639,17:POKE 640,13:POKE 198,10
5000 REM   ***     DATEN      ***
5010 DATA PROGRAMM-NR.1....,000100
5011 DATA PROGRAMM-NR.2....,000200
5012 DATA PROGRAMM-NR.3....,000300
5013 DATA PROGRAMM-NR.4....,000400
5014 DATA PROGRAMM-NR.5....,000100
5015 DATA PROGRAMM-NR.6....,000600
5016 DATA PROGRAMM-NR.7....,000700
5017 DATA PROGRAMM-NR.8....,000800
5018 DATA PROGRAMM-NR.9....,000900
5019 DATA PROGRAMM-NR.10...,001000
5020 DATA PROGRAMM-NR.11...,001100
5021 DATA PROGRAMM-NR.12...,001200
5022 DATA PROGRAMM-NR.13...,001300
5023 DATA PROGRAMM-NR.14...,001400
5024 DATA PROGRAMM-NR.15...,001500
5025 DATA PROGRAMM-NR.16...,001600
```

```
5026 DATA PROGRAMM-NR.17..,001700
5027 DATA PROGRAMM-NR.18..,001800
5028 DATA PROGRAMM-NR.19..,001900
5029 DATA PROGRAMM-NR.20..,002000
5030 DATA PROGRAMM-NR.21..,002100
5031 DATA PROGRAMM-NR.22..,002200
5032 DATA PROGRAMM-NR.23..,002300
5033 DATA PROGRAMM-NR.24..,002400
5034 DATA PROGRAMM-NR.25..,002500
5035 DATA PROGRAMM-NR.26..,002600
5036 DATA PROGRAMM-NR.27..,002700
5037 DATA PROGRAMM-NR.28..,002800
5038 DATA PROGRAMM-NR.29..,002900
5039 DATA PROGRAMM-NR.30..,003000
5040 DATA PROGRAMM-NR.31..,003100
5041 DATA PROGRAMM-NR.32..,003200
5042 DATA PROGRAMM-NR.33..,003300
5043 DATA PROGRAMM-NR.34..,003400
5044 DATA PROGRAMM-NR.35..,003500
5045 DATA PROGRAMM-NR.36..,003600
5046 DATA PROGRAMM-NR.37..,003700
5047 DATA PROGRAMM-NR.38..,003800
5048 DATA PROGRAMM-NR.39..,003900
5049 DATA PROGRAMM-NR.40..,004000
5050 DATA PROGRAMM-NR.41..,004100
5051 DATA PROGRAMM-NR.42..,004200
5052 DATA PROGRAMM-NR.43..,004300
5053 DATA PROGRAMM-NR.44..,004400
5054 DATA PROGRAMM-NR.45..,004500
5055 DATA PROGRAMM-NR.46..,004600
5056 DATA PROGRAMM-NR.47..,004700
5057 DATA PROGRAMM-NR.48..,004800
```

Dateiverwaltung für Apple

Neue Dateien eröffnen

Diese Funktion eröffnet beliebig viele Dateien auf der Diskette. Die Diskette muß allerdings das Programm enthalten.

Der Name der eröffneten Datei wird in der Liste dick gedruckt.

Daten hinzufügen

Diese Funktion ermöglicht es dem Benutzer Programme zu vorher errichteten Dateien hinzuzufügen. Zu diesem Zweck muß man zunächst den Dateinamen eingeben. Danach kann man zwischen vier verschiedenen Arten von Betriebssystemen wählen. Die Wahl erfolgt mittels Pfeiltasten und der Returntaste. Hat man sein Betriebssystem gewählt, so gelangt man in ein Menü, indem verschiedene Programmarten aufgeführt sind, wie z.B.: Actionspiele, Hires-Adventures, Musikprogramm usw. Hat man die Wahl beendet, fragt das Programm nach dem Namen und der Länge des Programms, das hinzugefügt werden soll. Zusätzlich kann man einen Index eingeben wie z.B. die Diskettenummer. Hat man die Eingabe beendet, fragt das Programm ob eine Anleitung vorhanden ist. Danach hat man die Möglichkeit mit Return die letzte Eingabe zu löschen (falls man sich vertippt hat), mit ESC die Funktion abzubrechen oder man kann mit der Leertaste die Eingabefunktion fortsetzen. Wenn man sich entschlossen hat, die Funktion abzubrechen, so beginnt das Programm, die eingegebenen Datensätze zu sortieren und springt danach wieder ins Hauptmenü zurück.

Dateien zusammenfügen

Diese Funktion ermöglicht es, einzelne Dateien in beliebiger Reihenfolge aneinanderzuhängen

Datei ausdrucken

Diese Funktion ermöglicht das Ausdrucken der einzelnen, vorher mit Funktion 3 zusammengefügten Dateien in einer Liste. Zunächst wird man nach dem Datum gefragt, dann sollte man sich vergewissern, daß man den Drukker eingestellt hat. Nachdem man Return gedrückt hat, bringt der Drucker die Liste auf Papier.
!!! Bei der Eingabe sollte man darauf achten, daß man in Zeile 4110 seine eigene Adresse eingegeben hat.
Das Programm ist auf einen FX-80+ und eine EPSON Printercard zugeschnitten. Es dürfte aber keine Schwierigkeiten bereiten, das Programm auf andere Drucker umzuschreiben.

Aus dem Programm aussteigen

Beendet das Programm

!!! Das Programm wird gelöscht.
!!! Die Daten, die zwar eingegeben, aber noch nicht gespeichert sind, werden gelöscht.

Daten löschen

Um Speicherplatz zu sparen wurde diese Funktion nicht definiert!
Sie kann aber nachträglich programmiert werden, die Einrichtungen sind vorhanden!

```
1  ONERR  GOTO 50000
2 D$ =  CHR$ (4)
5  GOSUB 60000
10  HOME
20 A$ = "----------------------------------------"
30 B$ = "                PROGRAMM-LISTE          "
40  PRINT A$;: INVERSE : PRINT B$;: NORMAL : PRINT A$;
50  POKE 34,3
60 C$(1,0) = "NEUE DATEI ERÖFFNEN"
70 C$(2,0) = "DATEN HINZUFUEGEN"
80 C$(3,0) = "DATEIEN ZUSAMMENFUEGEN"
90 C$(4,0) = "DATEN LOESCHEN"
100 C$(5,0) = "DATEI AUSDRUCKEN"
110 C$(6,0) = "AUS PROGRAMM AUSTEIGEN"
115 Q = 6:W = 2:U = 0
```

```
120 V = 7:H = 6:E = 5
130  GOSUB 15000
140  ON A GOTO 300,1000,2000,3000,4000,5000
300  HOME
310  VTAB 10: PRINT "INHALT DER DATEI - WIRD FETT AUSGEDRUCKT"
320  VTAB 15: HTAB 8: FOR X = 1 TO 33: PRINT "-";: NEXT
330  VTAB 14: INPUT "NAME : ";NAME$
340 I = 0
350  GOSUB 9000
360  GOSUB 21000
370  GOSUB 20000
380  POKE 34,0: HOME
390  CLEAR : GOTO 1
1000  HOME
1006 D$ =  CHR$ (4)
1010  VTAB 10: PRINT "ZU WELCHER DATEI WOLLEN SIE DATEN
                                    HINZUFUEGEN :"
1020  VTAB 13: HTAB 15: FOR U = 0 TO 25: PRINT "-";: NEXT
1030  VTAB 12: HTAB 14: INPUT " ";NAME$
1040  GOSUB 22000
1050 I = I + 1
1060  GOSUB 9000
1070  GOSUB 21000
1080  GOSUB 20000
1090  POKE 34,0: HOME
1100  CLEAR : GOTO 1
2000  HOME
2005  CLEAR
2007 P =  - 1:X = 1
2008 T = 1
2009  GOSUB 60000
2010  VTAB 11: PRINT "GEBEN SIE NAMEN DER ";X;". DATEI AN !"
2015 U = W:X = X + 1
2020  VTAB 15: PRINT "NAME :"
2030  VTAB 16: HTAB 8: FOR Y = 1 TO 31: PRINT "-";: NEXT
2050  VTAB 15: HTAB 8: INPUT " ";NAME$
2060  GOSUB 25000
2075  VTAB 22: HTAB 1: FLASH : PRINT "MIT 'ESC' STOPPEN, WEITER MIT ' ' .
    ": NORMAL
2080  VTAB 22: HTAB 21: GET B$: IF B$ = "" THEN 2080
2090  IF B$ =  CHR$ (27) THEN  POKE 34,0: HOME : GOTO 10
2100  IF B$ = " " THEN  HOME : GOTO 2010
2110  GOTO 2080
3000  HOME
3010  VTAB 12: PRINT "DIESE FUNKTION IST NOCH NICHT VORHANDEN !"
3020  CLEAR
3030  GOTO 1
4000  HOME
4010  IF T <  > 1 THEN  GOTO 18000
4015  FLASH
4016  GOSUB 55000
4017  HOME : VTAB 12: PRINT "GEBEN SIE DAS HEUTIGE DATUM EIN!": VTAB 14: HTAB
    1: INPUT "DATUM : ";DA$
4018  HOME
4020  VTAB 12: PRINT "HABEN SIE DEN DRUCKER ANGESTELLT ?"
4030  VTAB 14: PRINT "WENN JA, DANN DRUECKEN SIE 'RETURN' .";
4035  NORMAL
4040  GET A$: IF A$ = "" THEN 4040
4050  IF A$ =  CHR$ (13) THEN  GOTO 4100
4060  GOTO 4040
4100  PR# 1
4105  PRINT  CHR$ (27); CHR$ (77)
4110  HTAB 10: PRINT "JÖRGEN BARTSCH": HTAB 10: PRINT "HABICHTWEG 1": HTAB
    10: PRINT "6238 DIEDENBERGEN"
```

```
 4112  PRINT
 4115  HTAB 8: PRINT "TAUSCHE PROGRAMME UND ANLEITUNGEN ALLER ART !"; SPC(
       8);"LISTE VOM ";DA$
 4130  PRINT  CHR$ (27); CHR$ (77)
 4140  FOR A = 0 TO P
 4150  GOSUB 30000
 4155  PRINT  CHR$ (27); CHR$ (64)
 4160  PRINT  CHR$ (27); CHR$ (77)
 4170  GOSUB 16000
 4180  FOR Q = N TO I(A)
 4190  GOSUB 35000
 4200  HTAB 6: PRINT C$(D,1);
 4210  HTAB 17: PRINT C$(C,2);
 4220  HTAB 25: PRINT L$(Q);
 4230  HTAB 33: PRINT N$(Q);
 4235 E = 20 -  LEN (N$(Q))
 4240  PRINT  SPC( E);I$(Q);
 4245 E = 16 -  LEN (I$(Q))
 4250  PRINT  SPC( E);C$(B,3)
 4260  NEXT
 4265 N = I(A) + 1
 4267 I(A + 1) = I(A + 1) + I(A)
 4270  PRINT : PRINT
 4280  NEXT
 4290  PR# 0
 4300  POKE 34,0: HOME
 4310  GOTO 10
 5042 O(X) = I
 9000  REM  DATENABFRAGEROUTINE
 9003  HOME
 9004  POKE 34,0: VTAB 2: FLASH : PRINT I: NORMAL : POKE 34,3
 9005  VTAB 7: HTAB 4: PRINT "GEBEN SIE DAS BETRIEBSSYSTEM AN !"
 9010 C$(1,1) = "CP/M":C$(2,1) = "DOS 3.3":C$(3,1) = "PASCAL":C$(4,1) = "S
      ONSTIGES"
 9020 Q = 4:W = 2:U = 1
 9030 V = 11:H = 6:E = 9
 9040  GOSUB 15000
 9050 A1 = A:ZA = A
 9055 C =  LEN (C$(A,1))
 9060  HOME
 9070  VTAB 4: INVERSE : PRINT C$(A,1): NORMAL
 9080  POKE 34,4
10000  HOME
10010  VTAB 7: HTAB 27: FLASH : PRINT "PROGRAMMART": VTAB 8: HTAB 29: PRINT
       "WAEHLEN": NORMAL
10015 C$(1,2) = "ACTIONSPIEL":C$(2,2) = "ADVENTURE":C$(3,2) = "DENKSPIEL"
      :C$(4,2) = "HIRES-ADVENTURE"
10020 C$(5,2) = "KARTENSPIEL":C$(6,2) = "SIMULATION":C$(7,2) = "STRATEGIE
      SPIEL":C$(8,2) = "TEXT-ADVENTURE"
10030 C$(9,2) = "COPY-PROGRAMM":C$(10,2) = "DATENVERWALTUNG":C$(11,2) = "
      DISK-UTILITY"
10040 C$(12,2) = "GRAFIK-UTILITY":C$(13,2) = "MUSIKPROGRAMM":C$(14,2) = "
      PROGRAMMIERSPRACHE/-SYSTEM":C$(15,2) = "SONSTIGES":C$(16,2) = "TEXTV
      ERARBEITUNG"
10050 W = 1:Q = 16:U = 2
10060 V = 5:H = 6:E = 4
10070  GOSUB 15000
10080 A2 = A:ZA = ZA + (A * 10)
10090 C = C +  LEN (C$(A2,2)):D = 40 - C
10100  POKE 34,3: HOME : PRINT
10110  INVERSE : PRINT C$(A1,1); SPC( D);C$(A2,2): NORMAL
10120  POKE 34,5
11000  HOME
11010  VTAB 11: HTAB 7: FOR X = 1 TO 34: PRINT "-";: NEXT
```

```
11020  VTAB 10: PRINT "NAME :"
11030  VTAB 15: HTAB 9: FOR X = 1 TO 7: PRINT "-";: NEXT : PRINT  SPC( 25
    )
11040  VTAB 14: PRINT "LAENGE :"
11050  VTAB 19: HTAB 8: FOR X = 1 TO 33: PRINT "-";: NEXT
11060  VTAB 18: PRINT "INDEX :"
11070  VTAB 10: HTAB 7: INPUT " ";N$(I)
11072  VTAB 6: INVERSE : PRINT N$(I): NORMAL
11080  VTAB 14: HTAB 9: INPUT " ";L$(I)
11082 C =  LEN (N$(I)) +  LEN (L$(I)):D = 40 - C
11084  INVERSE : VTAB 6: PRINT N$(I); SPC( D);L$(I): NORMAL
11090  VTAB 18: HTAB 8: INPUT " ";I$(I)
11092  INVERSE : VTAB 7: PRINT I$(I): NORMAL
11094  POKE 34,7
11100  HOME
11110  FLASH : VTAB 9: HTAB 6: PRINT "IST BESCHREIBUNG VORHANDEN ?": NORMAL

11120 C$(1,3) = "JA":C$(2,3) = "NEIN":C$(3,3) = "UNNOETIG"
11130 Q = 3:W = 2:U = 3
11140 V = 12:H = 3:E = 10
11150  GOSUB 15000
11160 A3 = A
11165 ZA = ZA + (A * 1000):ZA$(I) =  STR$ (ZA)
11170  POKE 34,6
11180 C =  LEN (I$(I)) +  LEN (C$(A3,3)):D = 40 - C
11185  HOME
11190  INVERSE : VTAB 7: PRINT I$(I); SPC( D);C$(A3,3): NORMAL
11200  POKE 34,7
11210  HOME
11220  VTAB 12: PRINT "UM EINGABE ZU LOESCHEN 'CR' DRUECKEN !  "
11230  VTAB 14: PRINT "UM EINGABE ZU BEENDEN 'ESC' DRUECKEN !  "
11240  VTAB 16: PRINT "UM WEITEREINZUGEBEN 'LEERTASTE' DRUECK. "
11250  VTAB 2: GET B$: IF B$ = "" THEN 11250
11260  IF B$ =  CHR$ (13) THEN  POKE 34,3: HOME : GOTO 9000
11270  IF B$ =  CHR$ (32) THEN  POKE 34,3: HOME :I = I + 1: GOTO 9000
11280  IF B$ =  CHR$ (27) THEN  POKE 34,3: HOME : RETURN
11290  GOTO 11250
15000  REM  MENUESTEUERUNG
15010  FOR X = 1 TO Q

15020  VTAB V: HTAB H: PRINT C$(X,U)
15030 V = V + W
15040  NEXT
15050  FLASH : VTAB 22: PRINT "WAEHLEN SIE MIT PFEILTASTEN UND RETURN !"
15060 A = 1
15070 V = W * A + E
15080  INVERSE : VTAB V: HTAB H: PRINT C$(A,U): NORMAL
15090  VTAB 22: HTAB 30: GET B$: IF B$ = "" THEN 15090
15100  IF B$ =  CHR$ (13) THEN  RETURN
15110  IF B$ =  CHR$ (8) THEN 15150
15120  IF B$ =  CHR$ (21) THEN 15180
15130  GOTO 15090
15150  VTAB V: HTAB H: PRINT C$(A,U)
15160  IF A = 1 THEN A = Q + 1
15170 A = A - 1
15175  GOTO 15070
15180  VTAB V: HTAB H: PRINT C$(A,U)
15190  IF A = Q THEN A = 0
15200 A = A + 1
15210  GOTO 15070

16000  REM  UEBERSCHRIFTSROUTINE
16010  HTAB 5: PRINT "SYSTEM";: HTAB 16: PRINT "GRUPPE";: HTAB 24: PRINT
     "LAENGE";: HTAB 32: PRINT "PROGRAMMNAME";: PRINT  SPC( 16);: PRINT "
     INDEX";: PRINT  SPC( 9): PRINT "ANLEITUNG"
```

```
16020  HTAB 5: FOR X = 1 TO 79: PRINT "-";: NEXT
16030  PRINT
16040  RETURN
18000  HOME : FLASH : VTAB 11: PRINT "ES SIND KEINE DATEN VORHANDEN !
18010  VTAB 13: PRINT "ZUERST DATEN ZUSAMMENFUEGEN,"
18020  VTAB 15: PRINT " SIE DANN ERST AUSDRUCKEN !!!!";
18030  GET A$: IF A$ = "" THEN 18030
18040  POKE 34,0
18050  HOME
18060  GOTO 1

20000  REM ABSPEICHERUNGSROUTINE1
20010 D$ =  CHR$ (4)
20020  PRINT D$;"OPEN";NAME$
20030  PRINT D$;"WRITE";NAME$
20040  PRINT I: PRINT NAME$
20050  FOR X = 0 TO I
20060  PRINT L$(X): PRINT N$(X): PRINT I$(X): PRINT ZA$(X)
20080  NEXT
20090  PRINT D$;"CLOSE";NAME$
20100  RETURN
21000  REM  SORTIRROUTINE
21010  VTAB 11: PRINT "BITTE WARTEN, BIS DATEN GEORDNET SIND !"
21020  FOR X = 0 TO I
21025 F$(X) = N$(X) + "*" + L$(X) + "*" + I$(X) + "*" + ZA$(X)
21030  NEXT
21050  FOR U = I TO 1 STEP  - 1
21060 D = U - 1
21070  IF F$(D) > F$(U) THEN  GOSUB 21200
21080  NEXT
21090  IF S = 0 THEN 21300
21100 S = 0
21110  GOTO 21050
21200 K$ = F$(U):F$(U) = F$(D):F$(D) = K$
21210 S = 1
21220  RETURN
21300  FOR X = 0 TO I
21305 S = 1:V = 0:W = 0
21310  FOR O = 1 TO  LEN (F$(X))
21320  IF  MID$ (F$(X),O,1) = "*" THEN  ON S GOSUB 21400,21450,21500

21330  NEXT
21340  NEXT
21350  RETURN
21400 V = O - 1
21410 N$(X) =  LEFT$ (F$(X),V)
21420 V = O + 1
21430 S = S + 1
21440  RETURN
21450 W = O - V
21460 L$(X) =  MID$ (F$(X),V,W)
21470 V = O + 1
21480 S = S + 1
21490  RETURN
21500 W = O - V
21510 I$(X) =  MID$ (F$(X),V,W)
21520 V =  LEN (F$(X)) - V - W
21530 ZA$(X) =  RIGHT$ (F$(X),V)
21540  RETURN
22000  REM  DATEN EINLADE ROUTINE
22010  PRINT D$;"OPEN";NAME$
22020  PRINT D$;"READ";NAME$
22030  INPUT I: INPUT NAME$
22040  FOR Y = 0 TO I
22050  INPUT L$(Y): INPUT N$(Y): INPUT I$(Y): INPUT ZA$(Y)
```

```
22060  NEXT
22070  PRINT D$;"CLOSE";NAME$
22080  RETURN
25000  REM  DATENZUSAMMENFUEGROUTINE
25005 D$ =  CHR$ (4)
25007 P = P + 1
25010  PRINT D$;"OPEN";NAME$
25020  PRINT D$;"READ";NAME$
25030  INPUT I(P): INPUT NAME$(P)
25034 O = W + I(P)
25040  FOR Y = U TO O
25050  INPUT L$(Y): INPUT N$(Y): INPUT I$(Y): INPUT ZA$(Y)
25060  NEXT
25065 W = W + I(P) + 1
25070  PRINT D$;"CLOSE";NAME$
25080  RETURN

30000  REM  UEBERSCHRIFTSROUTINE
30010 B = (38 -  LEN (NAME$(A))) / 2
30015 B2 = B + .5
30020  PRINT  CHR$ (27); CHR$ (33); CHR$ (59)
30025  HTAB 3
30030  FOR X = 1 TO 39
30040  PRINT "*";
30050  NEXT
30052  PRINT "*"
30055  HTAB 3: PRINT "*"; SPC( 38);"*"
30060  HTAB 3: PRINT "*"; SPC( B);NAME$(A); SPC( B2);"*"
30065  HTAB 3: PRINT "*"; SPC( 38);"*"
30070  HTAB 3
30080  FOR R = 1 TO 39
30090  PRINT "*";
30100  NEXT
30110  PRINT "*"
30120  RETURN
35000  REM  AUFSCHLUESSELUNGSROUTINE
35020 B =  VAL ( LEFT$ (ZA$(Q),1))
35030 C =  VAL ( MID$ (ZA$(Q),2,2))
35040 D =  VAL ( RIGHT$ (ZA$(Q),1))

35050  RETURN
50000  REM  FEHLERERKENNUNGSROUTINE
50010 FE =  PEEK (222)
50020  IF FE = 5 THEN  PRINT D$;"CLOSE";NAME$: RESUME
50040  VTAB 12: PRINT " FEHLER NR.";FE;" IST AUFGETRETEN !"
50050  END
55000  REM  C$ AUFSCHLUESSLUNGSDATEI
55010 C$(1,1) = "CP/M":C$(2,1) = "DOS 3.3":C$(3,1) = "PASCAL":C$(4,1) = "
    SONSTIGES"
55020 C$(1,2) = "ACT":C$(2,2) = "ADV":C$(3,2) = "DSP":C$(4,2) = "HAD"

55030 C$(5,2) = "KAR":C$(6,2) = "SIM":C$(7,2) = "STR":C$(8,2) = "TAD"
55040 C$(9,2) = "COP":C$(10,2) = "DAT":C$(11,2) = "UTY"
55050 C$(12,2) = "GUT":C$(13,2) = "MUS":C$(14,2) = "SPR":C$(15,2) = "SON"
    :C$(16,2) = "TEX"
55060 C$(1,3) = "JA":C$(2,3) = "NEIN":C$(3,3) = ""
55070  RETURN
60000  REM  DIM ROUTINE
60010  DIM L$(700),N$(700),I$(700),ZA$(700)
60020  DIM F$(700): DIM C$(16,3)
60040  RETURN
61000  REM  ********************
61001  REM  *       (C)        *
61002  REM  * JUERGEN BARTSCH *
61003  REM  *                  *
61004  REM  ********************
```

Maschinensprache-Monitor für den TI-99

mit 32 k-Speichererweiterung

Will man in Maschinensprache programmieren, braucht man neben einem Assembler einen Maschinensprache-Monitor, mit dem man Speicherinhalte ausgeben, die Register anzeigen, Bytes suchen und Maschinenprogramme aufrufen lassen kann. Der folgende 99/4-Monitor bietet all dies und noch einiges mehr. Er läuft auf dem TI-99 mit Extended Basic und 32K-Speichererweiterung.

Der Monitor selbst ist in Extended Basic geschrieben, aber die wichtigsten Teile wie Speicherdumps und Suche nach Bytes sind in Maschinensprache; einerseits aus Geschwindigkeitsgründen und andererseits, weil viele Systemfunktionen in Basic gar nicht erreichbar sind.
Sobald Sie das Programmlisting abgetippt haben (achten Sie wegen der Prüfsummen auf die Zeilennummern!), können Sie das Programm starten. In Zeile 220 wird mit CALL CHECKSUM ein Unterprogramm aufgerufen, das die DATA-Zeilen auf ihre Richtigkeit prüft. Wird ein Fehler entdeckt, so stoppt das Programm und die Zeilennummer der fehlerhaften DATA-Zeile wird angegeben. Sind so alle DATAS geprüft, können Sie die Zeilen 220 und 2260 bis 2380 löschen und das Programm abspeichern. Ist die Diskette angeschlossen, Sie wollen das Programm aber auf Cassette abspeichern, kommt es wegen der Programmlänge zu Schwierigkeiten. Geben Sie in diesem Fall vor dem Abtippen CALL FILES(1) und NEW ein. Der Monitor selbst ist sehr einfach zu bedienen: Sie haben siebzehn Befehle zur Verfügung, die Sie aufrufen, indem Sie deren Anfangsbuchstaben eintippen. Drücken Sie z. B. „R" (ohne ENTER), werden die aktuellen Workspace-Register ausgegeben. Die folgende Tabelle zeigt alle Befehle mit Beschreibung:

A = „Assembler" – verknüpft das Programm mit einem Assembler, sofern ein solcher sich auf Cassette oder Diskette befindet (CS1 oder DSK1.ASSEMBLER).

B = „Breakpoint" – unterbricht den Monitor. Sie befinden sich wieder im Direktmodus und können Basic-Befehle verwenden. Der Monitor wird dann mit CONTINUE fortgesetzt.

D = „Decimal" – wandelt eine Dezimalzahl in eine Hexadezimal-Zahl um.
Beispiel:
>D 4096 = >1000

E = „Erase Screen" – löscht den Bildschirm.

F = „Find Word or Byte" – sucht ein Byte oder Word (2 Bytes) innerhalb vorgegebener Grenzen im Speicher – und zwar sowohl im normalen RAM/ROM als auch im VDP-RAM (Bildschirmspeicher) und im GROM. Dementsprechend können Sie der Startadresse ein „G" oder „V" voranstellen. Alle Fundstellen werden hexadezimal mit Adresse und Wert angegeben.
Beispiele:
>F WORD OR BYTE? W
START = 2000
END = 3FFF
VALUE = 02E0 sucht im Low Memory alle LWPI-Befehle (>02E0)
>F WORD OR BYTE? B
START = G0000
END = C000
VALUE = 00 sucht im GROM alle Null-Bytes

Mit ENTER können Sie die Suche beenden. Jede andere Taste führt dazu, daß das Listing angehalten wird; durch einen neuen Tastendruck geht es wieder weiter. Die Suche erfolgt übrigens in Maschinensprache mit der entsprechenden Geschwindigkeit.

G = „GROM Word Read" – liest ein Word aus dem GROM von der angegebenen Adresse. Mittels „+" und „–" können Sie den Speicher durchschreiten; mit ENTER können Sie dieses Unterprogramm verlassen.
Beispiel:
>G 0000 = AA02 –
FFFE = 0000 (Enter)

H = „Hexadezimal" – ist die Umkehrung des D-Befehls und wandelt eine Hexadezimalzahl in eine Dezimalzahl um:
>H A000 = -24576

I = „Iventory" – ist eine HELP-Funktion und listet auf dem Bildschirm alle Befehle auf.

J = „Jump" – ruft ein Maschinenprogramm mit seinem Namen auf, sofern dieser in der Namenstabelle steht:
>J ASMPRG

L = „Load" – lädt mittels CALL LOAD („Datei") ein Maschinenprogramm von der Diskette oder vom Cassettenrecorder:
>L DSK1.OBJECT 1

M = „Memory Read or Write" – funktioniert genauso wie das G-Kommando, nur daß Sie hier zusätzlich die Möglichkeit haben, an der Stelle, an der „+" oder „–" eingegeben wird, einen Hexwert einzutippen, der dann gepoked wird.
>M 24F4 = FFE0 +
24F6 = FE00 3FFF
24F8 = 03FF –
24F6 = 3FFF

Q = „Quit" – beendet die Arbeit mit dem Monitor, es erscheint die READY-Meldung.

R = „Registers Show" – holt sich den Workspace Pointer und zeigt daraufhin die sechzehn Workspace-Register an:
>R WP = 83 E0
R0=0001 R1=0010 R2=3F33
...
>S „Show Memory Dump" – ruft ein Maschinenprogramm auf, mit dem ein Memory-Dump von RAM, ROM, GROM und VDP-RAM angefertigt werden kann.
Beispiele:
>S 0000 00FF
>S G0000 01FF
>S V3FAC 3FFF
Die Tastenbedienung ist dieselbe wie bei der Find-Funktion.

T = „Text Fill" – ermöglicht es Ihnen, den RAM mit einem Text zu füllen:
>T 3000
DIES IST EIN PROBETEXT (Enter)

V = „VDP RAM Read/Write" – funktioniert genauso wie das M-Kommando, allerdings für den VDP-RAM:
>V 3FAC = 00FF +
3FAE = 3000 2000
3FB0 = FF35 (Enter)

X = „Execute Program" ruft ein Maschinensprache-Programm nicht mit den Namen auf, sondern mit der Startadresse und dem Workspace-Pointer:
>X
WP = 83E0 (Workspace-Pointer)
PC = 24FC (Programm-Zähler, Programm-Counter)
ACHTUNG: Steht an der angegebenen Adresse kein ausführbares Programm, so steigt der Computer aus und kann dann nur mit Reset zurückgeholt werden. Der Monitor muß dann natürlich neu geladen werden.

Die meisten Kommandos können Sie auch vorzeitig abbrechen, wenn Sie statt eines Wertes einfach „Enter" eingeben. Folgende Maschinenprogramme sind implementiert, die natürlich auch anderswo verwendet werden können:
- CALL LINK(„GETWP",X,WP) liefert den Workspace Pointer in WP.
- CALL LINK(„EXEC",WP,PC) ruft ein Maschinenprogramm nach der Adresse auf.
- CALL LINK(„MDUMP",Start,Ende) gibt auf dem Screen einen Memory Dump aus.
- CALL LINK(„VDUMP",Start,Ende) macht dasselbe für den VDP-RAM.
- CALL LINK(„GDUMP",Start,Ende) auch wieder ein Screen Dump, jedoch im GROM.
- CALL LINK(„VDEEK",Adr,Var) liest zwei Bytes (ein Word) aus dem VDP-RAM.
- CALL LINK(„MDEEK",Adr,Var) entspricht CALL PEEK(Adr,A,B) und Var=A*256+B.
- CALL LINK(„GDEEK",Adr,Var) macht dasselbe für den GROM.
- CALL LINK(„VDOKE",Adr,Wert) poked zwei Bytes in den VDP-RAM.
- CALL LINK(„MDOKE",Adr,Wert) poket zwei Bytes in den normalen RAM.
- CALL LINK(„MFIND",Start, Ende,-Wert,WB) sucht den RAM oder ROM nach einem Byte (WB=1) oder einem Word (WB=0) ab.
- CALL LINK(„GFIND",Start, Ende-,Wert,WB) und
- CALL LINK(VFIND",Start,Ende-,Wert,WB) tun dasselbe im GROM oder VDP-RAM.

Das Maschinenprogramm liegt übrigens zusammen mit dem Basic-Programm im High Memory ab –18912. Somit bleibt der gesamte Low Memory für Maschinenprogramme frei. Sie können natürlich auch absoluten Maschinencode ab >A000 laden, müssen aber darauf aufpassen, daß er dann nicht mit dem Monitor zusammenstößt.

Martin Kotulla

```
100 !  ********************
110 ! **********************
120 ! *                    *
130 ! *   99/4-MONITOR     *
140 ! *                    *
150 ! **********************
160 !  ********************
170 !
180 ! ----------------------
190 ! (C) Martin Kotulla,
200 ! ----------------------
210 !
220 CALL CHECKSUM ! Kann nach Probelauf zusammen mit 2260-2380 geloescht werden
230 CALL ASMPROG :: ON WARNING NEXT :: HX$="0123456789ABCDEF" :: ON ERROR 1410
240 DEF HEX$(X)=SEG$(HX$,INT(X/16)+1,1)&SEG$(HX$,X-INT(X/16)*16+1,1)
250 DEF HEXW$(X)=HEX$(INT(X/256))&HEX$(X-INT(X/256)*256)
260 DEF DEZ(X$)=(POS(HX$,SEG$(X$,1,1),1)-1)*16+POS(HX$,SEG$(X$,2,1),1)-1
270 DEF DEZW(X$)=DEZ(SEG$(X$,1,2))*256+DEZ(SEG$(X$,3,2))
280 DEF MS(X)=INT(X/256):: DEF LS(X)=X-INT(X/256)*256
290 DEF FU(X)=INT(X/2)*2
295 DEF F$(X$)=RPT$("0",4-LEN(X$))&X$
300 DEF PO(X)=X-65536*(X<0):: DEF NE(X)=X+65536*(X>32767)
310 DEF HPO$(X)=HEXW$(PO(X)):: DEF DNEG(X$)=NE(DEZW(F$(X$)))
320 DISPLAY AT(21,6)ERASE ALL:"** 99/4 MONITOR **" :: GOTO 350
330 PRINT
340 PRINT
350 CALL HCHAR(24,2,62)
360 CALL KEY(0,K,S):: IF K=13 OR K=32 THEN 340
370 IF S=0 OR K<65 OR K>90 THEN 360 ELSE K$=CHR$(K):: PRINT K$
380 ! ## A ##
390 IF K$<>"A" THEN 450
400 DISPLAY AT(23,5):"CS1 OR DSK1?"
410 ACCEPT AT(23,20)SIZE(-1):NA$
420 IF NA$="C" THEN RUN "CS1"
430 IF NA$="D" THEN RUN "DSK1.ASSEMBLER" ELSE 340
440 ! ## D ##
450 IF K$<>"D" THEN 490
```

```
460 ACCEPT AT(23,3)SIZE(6)VALIDATE(DIGIT,"-"):A
470 DISPLAY AT(23,14):"=  )";HPO$(A):: GOTO 340
480 ! ## E ##
490 IF K$="E" THEN 320
500 ! ## H ##
510 IF K$()"H" THEN 550
520 ACCEPT AT(23,3)SIZE(4)VALIDATE(HX$):A$
530 DISPLAY AT(23,14):USING "= ######":DNEG(A$):: GOTO 340
540 ! ## M/G/V ##
550 IF K$="M" OR K$="G" OR K$="V" THEN M$=K$ ELSE 660
560 ACCEPT AT(23,3)VALIDATE(HX$)SIZE(4):A$ :: A$=F$(A$)
570 ADR=DNEG(A$):: A$=HPO$(ADR):: PRINT
580 CALL LINK(M$&"DEEK",ADR,X):: X=PO(X)
590 PRINT A$;"=";HEXW$(X);"  ";CHR$(MS(X)AND 127);CHR$(LS(X)AND 127)
600 ACCEPT AT(23,17)SIZE(4)VALIDATE(HX$,"+-"):A$
610 IF A$="" THEN 340
620 IF A$="+" THEN ADR=ADR+2 ELSE IF A$="-" THEN ADR=ADR-2 ELSE 640
630 AD2=PO(ADR):: ADR=NE(AD2):: A$=HEXW$(AD2):: GOTO 580
640 CALL LINK(K$&"DOKE",ADR,DNEG(A$)):: ADR=ADR+2 :: GOTO 630
650 ! ## Q ##
660 IF K$="Q" THEN END
670 ! ## R ##
680 IF K$()"R" THEN 760 ELSE PRINT
690 RG=0 :: CALL LINK("GETWP",X,C):: DISPLAY AT(22,9):"WP=";HPO$(C)
700 FOR I=C TO C+30 STEP 2 :: CALL PEEK(I,A,B)
710 RG$=CHR$(RG+48-7*(RG)9))
720 PRINT "R";RG$;"=";HEX$(A);HEX$(B);" ";
730 RG=RG+1 :: IF INT(RG/3)=RG/3 THEN PRINT
740 NEXT I :: GOTO 330
750 ! ## J ##
760 IF K$()"J" THEN 810
770 ACCEPT AT(23,3)SIZE(6):A$
780 IF A$()"" THEN PRINT :: CALL LINK(A$)
790 GOTO 340
800 ! ## L ##
810 IF K$()"L" THEN 840
820 ACCEPT AT(23,3):A$ :: CALL LOAD(A$):: GOTO 340
830 ! ## T ##
840 IF K$()"T" THEN 890
850 ACCEPT AT(23,3)VALIDATE(HX$):A$ :: ADR=DNEG(A$):: PRINT : :
860 CALL KEY(0,K,S):: IF S(1 THEN 860 ELSE IF K=13 THEN 330
870 PRINT CHR$(K);:: CALL LOAD(ADR,K):: ADR=NE(ADR+1):: GOTO 860
880 ! ## X ##
890 IF K$()"X" THEN 940
900 PRINT :"WP=" :: ACCEPT AT(23,4):WP$ :: IF WP$="" THEN 340
910 PRINT "PC=" :: ACCEPT AT(23,4):PC$ :: IF PC$="" THEN 340
920 CALL LINK("EXEC",DNEG(WP$),DNEG(PC$)):: GOTO 340
930 ! ## S ##
940 IF K$()"S" THEN 1030
950 ACCEPT AT(23,3)SIZE(5)VALIDATE(HX$,"GV"):A$
960 IF A$="" THEN 340 ELSE AS=ASC(A$)
970 IF AS=86 OR AS=71 THEN NA$=CHR$(AS):: A$=SEG$(A$,2,4)ELSE NA$="M"
980 IF A$="" THEN 340 ELSE ST=DNEG(SEG$(A$,LEN(A$)-3,4))
990 ACCEPT AT(23,9)SIZE(4)VALIDATE(HX$):A$
1000 IF A$="" THEN 340 ELSE EN=DNEG(A$)
1010 PRINT :: CALL LINK(NA$&"DUMP",ST,EN):: GOTO 340
1020 ! ## F ##
1030 IF K$()"F" THEN 1180
1040 DISPLAY AT(23,3):"WORD OR BYTE?"
1050 ACCEPT AT(23,17)SIZE(1)VALIDATE("WB"):WB$
1060 IF WB$="B" THEN SI=2 :: WB=1 :: GOTO 1080
1070 IF WB$="W" THEN SI=4 :: WB=0 ELSE 340
1080 PRINT :"START =" :: ACCEPT AT(23,9)SIZE(5)VALIDATE(HX$,"GV"):ST$
1090 IF ST$="" THEN 340 ELSE AS=ASC(ST$)
1100 IF AS=86 OR AS=71 THEN NA$=CHR$(AS):: ST$=SEG$(ST$,2,4)ELSE NA$="M"
1110 ST=DNEG(SEG$(ST$,LEN(F$(ST$))-3,4))
```

```
1120 PRINT "END   =" :: ACCEPT AT(23,9)SIZE(4)VALIDATE(HX$):EN$
1130 IF EN$="" THEN 340 ELSE EN=DNEG(EN$)
1140 PRINT "VALUE =" :: ACCEPT AT(23,9)SIZE(SI)VALIDATE(HX$):VA$
1150 IF VA$="" THEN 340 ELSE VA=DNEG(VA$)
1160 PRINT :: CALL LINK(NA$&"FIND",ST,EN,VA,WB):: GOTO 340
1170 ! ## I ##
1180 IF K$()"I" THEN 1370 ELSE PRINT
1190 PRINT "A = RUN ASSEMBLER"
1200 PRINT "B = BREAKPOINT IN MONITOR"
1210 PRINT "D = DECIMAL TO HEXADECIMAL"
1220 PRINT "E = ERASE SCREEN"
1230 PRINT "F = FIND WORD OR BYTE"
1240 PRINT "G = READ WORD FROM GROM"
1250 PRINT "H = HEXADECIMAL TO DECIMAL"
1260 PRINT "I = INVENTORY OF COMMANDS"
1270 PRINT "J = JUMP INTO PROGRAM"
1280 PRINT "L = LOAD PROGRAM FROM DISK"
1290 PRINT "M = MEMORY READ OR WRITE"
1300 PRINT "Q = QUIT MONITOR"
1310 PRINT "R = SHOW WORKSPACE REGISTERS"
1320 PRINT "S = SHOW MEMORY DUMP"
1330 PRINT "T = FILL MEMORY WITH TEXT"
1340 PRINT "V = VDP RAM READ OR WRITE"
1350 PRINT "X = EXECUTE PROGRAM" :: GOTO 340
1360 ! ## B ##
1370 IF K$()"B" THEN 1390 ELSE BREAK
1380 GOTO 340
1390 CALL SOUND(150,220,0):: PRINT :"* ILLEGAL COMMAND" :: GOTO 340
1400 ! ********************
1410 CALL ERR(EC,ET,ES,EL):: ON ERROR 1410 :: CALL SOUND(150,220,0)
1420 IF EC=130 THEN PRINT "* I/O ERROR" :: RETURN 340
1430 IF EC=135 THEN PRINT "* SUBPROGRAM NOT FOUND" :: RETURN 340
1440 IF EC=74 THEN PRINT "* INCORRECT ARGUMENT LIST" :: RETURN 340
1450 PRINT :"* ERROR #";STR$(EC);" IN";EL :: CALL LOAD(-31932,0)
1460 ! ********************
1470 SUB ASMPROG :: RESTORE
1480 DISPLAY AT(8,9)BEEP ERASE ALL:"99/4-MONITOR"
1490 CALL HCHAR(10,3,42,28)
1500 DISPLAY AT(15,9):"BITTE WARTEN": :TAB(8);"ICH LESE DATEN"
1510 DISPLAY AT(24,3):"(C) MARTIN KOTULLA 1984"
1520 CALL PEEK(8198,A):: IF A()170 THEN CALL INIT
1530 FOR I=-18912 TO -18012 STEP 10 :: READ A,B,C,D,E,F,G,H,J,K
1540 CALL LOAD(I,A,B,C,D,E,F,G,H,J,K):: NEXT I
1550 CALL PEEK(8196,A,B):: C=A*256+B-112
1560 CALL LOAD(8196,INT(C/256),C-INT(C/256)*256)
1570 RESTORE 2180
1580 FOR I=C TO C+112 STEP 8 :: READ A,B,C,D,E,F,G,H
1590 CALL LOAD(I,A,B,C,D,E,F,G,H):: NEXT I
1600 !
1610 DATA 048,049,050,051,052,053,054,055,056,057,065,066,067,068,069,070
1620 DATA 032,000,013,000,127,030,032,032,032,032,061,032,032,032,032,032
1630 DATA 032,032,032,032,032,032,032,032,032,032,032,032,032,032,032,032
1640 DATA 032,032,016,000,010,142,000,000,000,000,002,160,200,000,131,074
1650 DATA 002,224,182,000,006,160,185,122,004,096,185,150,002,224,182,000
1660 DATA 002,001,000,001,006,160,185,106,200,032,131,074,182,132,006,160
1670 DATA 185,106,002,224,131,224,192,032,131,074,006,144,004,096,185,150
1680 DATA 002,224,182,000,006,160,185,076,002,001,000,001,006,160,185,106
1690 DATA 192,096,131,074,006,160,185,092,216,032,152,000,131,074,216,032
1700 DATA 152,000,131,075,006,160,185,122,004,096,185,140,004,096,185,150
1710 DATA 002,224,182,000,002,001,000,001,006,160,185,106,192,224,131,074
1720 DATA 006,160,035,202,200,001,131,074,006,160,185,122,004,096,185,150
1730 DATA 002,224,182,000,002,001,000,001,006,160,185,106,193,032,131,074
1740 DATA 006,160,185,106,192,096,131,074,006,160,035,230,004,096,185,150
1750 DATA 002,224,182,000,002,001,000,001,006,160,185,106,192,160,131,074
```

```
1760 DATA 200,018,131,074,006,160,185,122,004,096,185,150,002,224,182,000
1770 DATA 002,001,000,001,006,160,185,106,193,032,131,074,006,160,185,106
1780 DATA 197,032,131,074,004,096,185,150,002,224,182,000,002,012,184,212
1790 DATA 016,009,002,224,182,000,002,012,184,238,016,004,002,224,182,000
1800 DATA 002,012,184,216,002,001,000,004,006,160,185,106,193,096,131,074
1810 DATA 019,006,200,032,182,084,183,152,007,032,182,086,016,005,200,032
1820 DATA 182,082,183,152,004,224,182,086,006,160,185,076,002,001,000,001
1830 DATA 006,160,185,106,193,096,131,074,006,160,185,106,193,160,131,074
1840 DATA 006,160,185,106,195,160,131,074,016,000,200,032,182,086,182,086
1850 DATA 022,021,006,156,130,078,019,004,129,133,018,251,004,096,185,140
1860 DATA 194,069,006,073,002,004,182,054,006,160,184,166,194,078,006,160
1870 DATA 184,166,002,003,000,009,006,160,185,000,016,235,006,156,004,224
1880 DATA 182,088,147,137,019,011,006,201,147,137,022,004,005,133,007,032
1890 DATA 182,088,016,004,129,133,018,242,004,096,185,140,194,069,006,073
1900 DATA 002,004,182,054,006,160,184,166,194,078,006,160,184,166,002,003
1910 DATA 000,007,006,160,185,000,200,032,182,088,182,088,019,223,006,005
1920 DATA 016,221,002,224,182,000,002,012,184,212,016,009,002,224,182,000
1930 DATA 002,012,184,238,016,004,002,224,182,000,002,012,184,216,006,160
1940 DATA 185,076,002,001,000,001,006,160,185,106,193,096,131,074,006,160
1950 DATA 185,106,193,160,131,074,002,004,182,054,194,069,006,160,184,166
1960 DATA 006,160,184,116,200,009,182,076,006,160,184,116,200,009,182,078
1970 DATA 006,160,184,116,200,009,182,080,006,160,184,252,129,133,018,235
1980 DATA 004,096,185,140,194,139,006,156,006,160,184,166,002,073,127,127
1990 DATA 152,009,182,048,017,003,152,009,182,052,017,002,210,096,182,053
2000 DATA 006,201,152,009,182,048,017,003,152,009,182,052,017,002,210,096
2010 DATA 182,053,006,201,004,090,192,137,009,194,221,034,182,032,192,137
2020 DATA 002,066,015,000,009,130,221,034,182,032,192,137,002,066,000,240
2030 DATA 009,066,221,034,182,032,192,137,002,066,000,015,221,034,182,032
2040 DATA 005,132,004,091,194,117,004,091,195,075,192,069,006,160,185,092
2050 DATA 210,096,152,000,216,032,152,000,182,019,005,197,004,093,195,075
2060 DATA 192,197,006,160,035,202,194,065,005,197,004,093,002,003,000,028
2070 DATA 002,000,002,226,002,002,182,054,208,114,002,033,096,000,004,032
2080 DATA 032,032,005,128,006,003,022,248,004,032,032,024,000,038,004,032
2090 DATA 032,028,193,224,131,124,033,224,182,048,022,015,152,032,131,117
2100 DATA 182,050,019,044,004,032,032,028,193,224,131,124,033,224,182,048
2110 DATA 022,249,152,032,131,117,182,050,019,033,004,091,210,032,152,002
2120 DATA 006,200,210,032,152,002,006,200,006,008,004,091,216,001,156,002
2130 DATA 006,193,216,001,156,002,006,193,004,091,004,192,004,032,032,012
2140 DATA 004,032,032,024,018,184,005,129,004,091,004,192,002,001,000,002
2150 DATA 004,032,032,024,000,032,004,032,032,008,004,091,216,008,156,002
2160 DATA 006,200,216,008,156,002,216,032,000,002,131,124,002,224,131,224
2170 DATA 004,096,000,112,000,001,002,003,004,000,000,000,000,000,000,000
2180 DATA 071,070,073,078,068,032,183,076,086,070,073,078,068,032,183,066
2190 DATA 077,070,073,078,068,032,183,056,071,068,079,075,069,032,182,188
2200 DATA 077,068,079,075,069,032,183,028,086,068,079,075,069,032,182,224
2210 DATA 071,068,069,069,075,032,182,144,077,068,069,069,075,032,183,000
2220 DATA 086,068,069,069,075,032,182,192,071,068,085,077,080,032,184,038
2230 DATA 086,068,085,077,080,032,184,028,077,068,085,077,080,032,184,018
2240 DATA 069,088,069,067,032,032,182,108,071,069,084,087,080,032,182,090
2250 SUBEND
2260 ! **********************
2270 SUB CHECKSUM :: RESTORE 2330 :: DIM CHK(64)
2280 PRINT "BITTE WARTEN! ICH PRUEFE":"DIE DATA-ZEILEN!": :
2290 FOR I=1 TO 64 :: READ CHK(I):: NEXT I :: RESTORE 1610
2300 FOR I=1 TO 64 :: SUM=0 :: FOR J=1 TO 8 :: READ A,B :: SUM=SUM+A+B :: NEXT J
2310 IF SUM=CHK(I)THEN DISPLAY BEEP:I*10+1530;"KORREKT!" ELSE CALL SOUND(150,220
,0):: PRINT " FEHLER IN";I*10+1530 :: STOP
2320 NEXT I :: PRINT "DIE DATA'S SIND KORREKT!"
2330 DATA 930,551,512,799,1724,1378,1886,1296,1789,1691,1490,1717,1299,1816,1426
2340 DATA 1739,1348,1687,1297,1372,1418,1526,1966,1801,1392,1538,1356,1278,1579
2350 DATA 1551,1383,1488,1444,1382,1870,1868,2054,1765,1366,1338,1866,1328,1425
2360 DATA 1612,1628,1383,959,642,1688,1550,1478,1292,1144,724,677,1674,222,1307
2370 DATA 1401,1426,1283,1408,1261,1342
2380 SUBEND
```

Hardcopy für TI-99

Beginn des Hardcopyunterprogramms: Dimensionierung der sogenannten binären Überlaufstrings für 32 Spalten. Kanal #1 wird auf hochauflösende Grafik eingestellt.
Binärstring (Zahlen von 0 bis 16 in Binärform aneinandergereiht) wird definiert.
Hexadezimalstring (Zahlen von 0 bis 16 in Hexadezimalform aneinandergereiht) wird definiert.
String aus 64 Nullen und ein Zeilenvorschub für hochauflösende Grafik werden definiert.
Bildschirmzeilenabtastschleife (von 1 bis 24).
Bildschirmspaltenabtastschleife (von 1 bis 32).
Der ASCII-Code einer der 768 Bildschirmpositionen wird ermittelt und der binäre Zeichencodestring wird gelöscht.
Falls der ASCII-Code nicht zwischen 32 und 143 liegt, ist der binäre Zeichencodestring gleich dem 64-Nullenstring und es geht in Zeile 30030 weiter.
Der hexadez. Zeichencodestring vom ermittelten ASCII-Code wird abgefragt. Falls der hexadez. Zeichencodestring aus 16 Nullen besteht, ist der bin. Zeichencodestring gleich dem 64-Nullenstring und es geht weiter in Zeile 30030.
Der 16stellige, hexadez. Zeichencodestring wird in einen 64stelligen, binären Zeichencodestring transformiert.
Ein bin. Überlaufstring wird mit dem bin. Zeichencodestring verknüpft. Das Ganze ist jetzt der neue bin. Zeichencodestring.
Der Teil des bin. Zeichencodestrings, der nach der 56. Position beginnt, wird ermittelt. Dieser Teil ist ein neuer bin. Überlaufstring.
Die ersten 56 Positionen des bin. Zeichencodestrings werden an das Druckunterprogramm übergeben.
Die Spaltenabtastschleife wird um eins erhöht, d. h. das nächste Bildschirmzeichen wird umgewandelt.
Zeilenvorschub auf dem Drucker.
Falls der bin. Überlaufstring der 32. Spalte 56 Positionen enthält, wird er als ganze Zeile auf dem Drucker ausgegeben und gelöscht, anschließend ein Zeilenvorschub, ansonsten erfolgt ein Sprung in Zeile 30050.
Die Zeilenabtastschleife wird um eins erhöht, d. h. die nächste Zeile auf dem Bildschirm wird nun spaltenweise abgetastet und transformiert. Den letzten bin. Überlaufstrings, die nicht aus 56 Positionen bestehen, werden einige Nullen hinzugefügt. Dann werden sie als ganze Zeile einzeln dem Druckunterprogramm übergeben.
Der letzte Zeilenvorschub auf dem Drucker.
Kanal #1 wird geschlossen, die Hardcopy ist erstellt.
Die langen Strings werden gelöscht.
Das Hardcopyprogramm ist beendet.
Beginn des Druckunterprogramms, der Parameter ist entweder der 56 Positionen beinhaltende bin. Zeichencodestring oder der bin. Überlaufstring. Matrixspaltenschleife für ein Zeichen, das aus einer 8 x 7 Matrix besteht, wird begonnen (von 1 bis 8). Die Variable für hochauflösende Druckergrafik wird auf Leerdruck zurückgestellt.
Matrixzeilenschleife (von 0 bis 6).
Zur Druckvariablen wird ein umgewandelter bin. Übergabestring addiert.
Nächste Matrixzeile.
Eine Spalte der Matrix wird an den Drucker übergeben.
Die nächste Matrixspalte wird berechnet.
Ende des Druckunterprogramms.

```
20000 ! ####################
20005 ! #                  #
20010 ! #                  #
20015 ! #                  #
20020 ! #  BASIC-HARDCOPY  #
20030 ! #   LONG-VERSION   #
20040 ! #   (C)BY KABOSO   #
20050 ! # KARSTEN  BOGUSCH #
20060 ! #     SOFTWARE     #
20070 ! #                  #
20080 ! #                  #
20090 ! #                  #
20100 ! ####################
20110 !
20120 !
20130 !
30000 SUB HARD
30001 DIM U$(32)
30002 OPEN #1:"RS232.DA=8.CR.LF" :: PRINT #1:CHR$(8);
30003 B$="0000000100100011010001010111001111000100110101011110011011110111 1"
30004 H$="0123456789ABCDEF"
30005 T$=RPT$("0",64):: C$=CHR$(13)&CHR$(10)
30010 FOR Z=1 TO 24
30011 FOR S=1 TO 32
30012 CALL GCHAR(Z,S,Q):: Z$=""
30013 IF Q<32 OR Q>143 THEN Z$=T$ :: GOTO 30030
```

```
30014 CALL CHARPAT(Q,Q$)
30015 IF Q$=SEG$(T$,1,16)THEN Z$=T$ :: GOTO 30030
30020 FOR I=1 TO 16
30021 Z$=Z$&SEG$(B$,1+4*(POS(H$,SEG$(Q$,I,1),1)-1),4)
30022 NEXT I
30030 Z$=U$(S)&Z$
30031 U$(S)=SEG$(Z$,57,LEN(Z$)-56)
30032 CALL D(SEG$(Z$,1,56))
30033 NEXT S
30034 PRINT #1:C$
30035 IF LEN(U$(32))<>56 THEN 30050
30040 FOR I=1 TO 32 :: CALL D(U$(I)):: U$(I)="" :: NEXT I
30041 PRINT #1:C$;
30050 NEXT Z
30051 FOR I=1 TO 32 :: CALL D(U$(I)&SEG$(T$,1,32)):: U$(I)="" :: NEXT I
30052 PRINT #1:C$;
30053 CLOSE #1
30054 B$,H$,T$=""
30055 SUBEND
30070 SUB D(G$)
30071 FOR N=1 TO 8 :: K=128
30072 FOR M=0 TO 6
30073 K=K+2^M*-(VAL(SEG$(G$,8*M+N,1))>0)
30074 NEXT M
30075 PRINT #1:CHR$(K);
30076 NEXT N
30077 SUBEND
```

Die geschichtliche Entwicklung der Datenverarbeitung im Zeitraffer

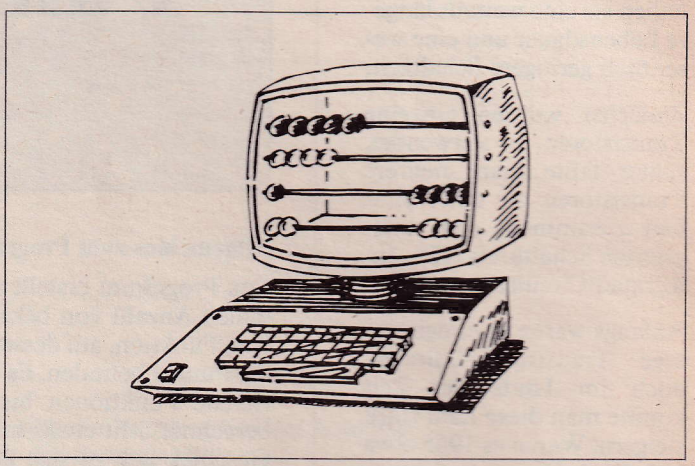
Von Oliver Steinmeier und Klaus Schneider

DIE ANFÄNGE DER DATENVERARBEITUNG UND MECHANISCHEN RECHENMASCHINEN

Schon lange vor den ersten Computern nutzten die Menschen Maschinen als Hilfsmittel zum Rechnen.
So wurde vor ca. 4000 Jahren der Abakus erfunden. Eingesetzt wurde er in Japan und China, aber auch die alten Römer haben ihn benutzt.
Auch heute noch erreichen die Japaner mit dem Abakus sehr hohe Rechengeschwindigkeiten. Experten sind in der Lage, 50 Divisionsaufgaben mit 5 bis 7stelligen Zahlen in etwa 90 Sekunden zu lösen. Sie addieren 15 zehnstellige Zahlen in nur 20 Sekunden.
Weitere Schritte zum modernen Rechner machte man ab dem 17. Jahrhundert.
Schon 1620 fiel dem englischen Mathematiker GUNTER auf, daß sich mit Logarithmen schneller rechnen läßt, wenn man auf einem Maßstab die logarithmischen Werte als Strecken vom Ausgangspunkt aufträgt. Die Genauigkeit ist dabei vom Maßstab abhängig. Der Rechenstab ist somit eine für jede Zahl einstellbare Logarithmentafel.
1623/24 schrieb WILHELM SCHICKARD eine Arbeit über Rechenmaschinen. Eine 1950 nach seinen Aufzeichnungen gebaute Vierspeziesrechenmaschine ist voll funktionsfähig. Sie ist heute im Deutschen Museum in München ausgestellt.
BLAISE PASCAL ließ im Jahre 1641 eine Additions- und Subtraktionsmaschine bauen, um Steuern leichter und schneller berechnen zu können.
Da SCHICKARDS Rechenmaschine in den Wirren des Dreißigjährigen Krieges verloren ging, galt PASCALS Maschine, obwohl sie primitiver war, lange Zeit als erste auf ihrem Gebiet.
Eine Maschine ähnlich der Pascals erfand G. W. LEIBNIZ. Dieses Gerät war jedoch einfacher konstruiert, da er das Prinzip der Staffelwalzen erfand. Außerdem erkannte Leibniz schon die Möglichkeiten der Dualzahlen in der Datenverarbeitung.
Das Lochkartenprinzip wurde in der Wirtschaft zuerst von J. M. JACQUARD benutzt. Er vervollständigte 1805 den mechanischen Musterwebstuhl durch Programmierung mit Lochkarten.
CHARLES BABBAGE entwickelte 1823 ebenfalls auf der Basis von Lochkarten eine Differenziermaschine, die zur Berechnung von Funktionen diente. Trotz großer Fortschrittlichkeit hatte er nur geringen Erfolg, da die Mechanik zu aufwendig war.
GEORGE BOOLE – nach ihm wurde die Boolesche Algebra benannt – steuerte die wichtigsten Grundlagen zur logischen Algebra bei. (0 für falsch, 1 für richtig).
HERMANN HOLLERITH (deutscher Vater) setzte bei der amerikanischen Volkszählung 1890 erstmals Lochkarten als Informationsträger im großen Stil ein. Dieses Prinzip fand auf Grund der großen Zeitersparnis auch in Österreich und Rußland Verbreitung. Vor der Lochkartenverarbeitung benötigten 500 Mitarbeiter 7 Jahre zur Auswertung der im Abstand von 5 Jahren stattfindenden Zählung, nun schaffte man diese Arbeit in etwa 7 Monaten und brauchte nur noch 43 Arbeitskräfte.

RECHNER MIT RELAISTECHNIK

1941 konstruierte der deutsche Forscher und Techniker KONRAD ZUSE den bekannten „Rechner" Z 3. Seine Maschine bestand aus ca. 2600 Relais; gesteuert wurde sie durch gelochten Kinofilmstreifen. Z 3 entwickelte



eine Rechengeschwindigkeit von 15 – 20 arithmetischen Operationen pro Sekunde. Eine Multiplikation dauerte etwa 4 – 5 Sekunden.
Z 3 konnte 64 Zahlen mit je 22 Dualstellen speichern und verarbeiten. Für den Speicher benötigte ZUSE 2000 Relais, 600 wurden im Rechenwerk eingesetzt.
Neben den vier Grundrechenarten waren das Wurzelziehen und die Multiplikation mit 2, ½, 10, $^1/_{10}$, -1 eingebaut. Zuse Z 3 kostete etwa 25 000 Reichsmark.

In den USA wurde 1944 MARK 1 durch HOWARD H. AIKEN mit Unterstützung von IBM an der Haward University in Cambridge/Massachusetts fertiggestellt. MARK 1 bestand aus 13 000 Relais und kostete damals 400 000 Dollar. Er wurde durch die Programmierung der Einzelmaschinen mittels Stecktafeln gesteuert.
Dieser Riesencomputer hatte eine Länge von 16 Metern und wog 35 Tonnen. Für eine Multiplikation benötigte er etwa 6 Sekunden.
AIKEN baute MARK 2, 3 und 4, bei denen erstmals Magnettrommeln und Magnetbänder als Speichermedium eingesetzt wurden.

COMPUTER MIT RÖHRENTECHNIK

In den Jahren 1943 – 46 wurde von Mitarbeitern der Moore-School of Electrical Engeneering der legendäre ENIAC (Elektronic Numerical Integrator and Computer) gebaut.
Er bestand aus 18 000 Elektronenröhren, die zusammen etwa 200 kW verbrauchten.
ENIAC stand auf einer Fläche von 135 Quadratmetern und kostete 10 Mio. Dollar.
Da Elektronenröhren eine geringe Lebensdauer haben, mußte bei 18 000 Röhren fast ständig nach Fehlern und fehlerhaften Röhren gesucht werden.
ENIAC war der erste Elektronenröhrenrechner und benötigte zur Multiplikation zweier 10stelliger Dezimalzahlen nur 2,8 Millisekunden.
Eingesetzt wurde er vorwiegend für ballistische Berechnungen.
Erst 1955 wurde er außer Dienst gestellt und steht heute im Nationalmuseum in Washington.

TRANSISTORTECHNIK IN DER DATENVERARBEITUNG

Nachdem 1954 der erste Silizium-Transistor gebaut wurde, konnte man nun die Computer statt der Elektronenröhren mit Transistoren bauen.

Transistoren haben gegenüber den Röhren viele Vorteile. Sie sind viel kleiner und haben eine geringere Stromaufnahme. Außerdem haben sie eine weitaus längere Lebensdauer und eine wesentlich geringere Schaltzeit.

Zunächst wurden einzelne Transistoren verwendet, später faßte man mehrere Transistoren zu einer Einheit zusammen, dem integrierten Schaltkreis (IC – Integrated Circuit).

Anfangs waren es einige wenige Transistorfunktionen, doch im Laufe der Zeit konnte man diese Zahl stark steigern. Waren es 1965 etwa 100 Funktionen, so konnte man 1970 schon 1000 Bauteile zusammenfassen (heute sind es mehrere 100 000).

CHIPS IN DER DATENVERARBEITUNG

1970 kam es zur Entwicklung des ersten Ein-Chip-Mikroprozessors durch den Amerikaner GARY BOONE von Texas Instruments.

Heute gibt es eine Vielzahl verschiedener Prozessoren mit unterschiedlichen Bus-Breiten (8, 16, 32 Bit). In den heutigen Home- und Personalcomputern werden hauptsächlich folgende Mikro-Prozessoren verwendet: 6502, 6809, Z-80A, 8088, 68000.

Es ist fast selbstverständlich, daß Chips heute mit Hilfe von Computern entwickelt und produziert werden.
Die Einsatzmöglichkeiten von Mikroprozessoren sind vielfältig. Sie rechnen und vergleichen, steuern und regeln, treffen logische Entscheidungen, sammeln und verarbeiten Daten.

So erfolgt die Abarbeitung des Waschprogramms einer Waschmaschine mit Hilfe eines solchen Mikroprozessors.

Die Entwicklung ist in diesem Bereich keineswegs abgeschlossen, es werden immer leistungsfähigere und schnellere Computer und Mikroprozessoren gebaut.

Funktionsanalyse für TI-99

Allgemeines zum Programm

Das Programm erstellt aus einer minimalen Anzahl von bekannten Punkten eine Funktion, auf dessen Graphen sich die Punkte befinden. Es werden ganzrationale Funktionen bis zum 5. Grad berechnet. Mit etwas mathematischem Geschick läßt es sich auch auf andere Funktionsstufen umstellen.
Das Programm ist weitgehend Menuegesteuert, zudem ist es abbruchsicher gestaltet mit den Anweisungen: ON WARNING NEXT und ON ERROR 1690.
Es belegt samt seinen Variablen nahezu den gesamten Speicherplatz des TI. Weiterhin benötigen Sie für dieses Programm mindestens das Extended-Basic-Modul, ein Drucker ist empfehlenswert, da Sie alle errechneten Werte zusammenfassend ausdrucken können. Das Programm spricht den Matrixdrucker GP-100 A von Seikosha an. Profis sollte es keine Schwierigkeiten bereiten, das Programm für andere Drucker umzuschreiben.

Leistungen:

Falls Sie dem Rechner eine bereits bekannte Funktion eingegeben haben, erstellt er ihnen eine komplette Kurvendiskussion. Er berechnet Y-Achsenschnittpunkt, Nullstellen, Extrem-Punkte und Wendepunkte. Der Computer berechnet die Fläche unter der Kurve, innerhalb eines variablen X-Achsenintervalls. Eine graphische Darstellung ist ebenfalls integriert, wobei Sie die Abzissen- und Ordinatenbereiche selbst festlegen.

Mit wenigen Punkten zur Funktion

Sollten nur einige Punkte des Graphen bekannt sein, muß das Programm zuerst die Koeffizienten der gesuchten Funktion berechnen. Anm.: Ohne den Computer würde dieses einen gewaltigen Zeitaufwand von vielen Stunden darstellen. Um die Anzahl der nötigen Punkte herauszufinden, addieren Sie zum Grad der Funktion eins.

Beispielberechnung:

Es handelt sich um eine Polynomfunktion 5. Grades. Sie benötigen hier also genau sechs Punkte.
Die Punkte seien [P (x:y)]:
P1(-51:100) P2(-3:-10) P3(-2:-2) P4(2:10) P5(4:-90) P6(7:-10)
Nun wissen Sie [f(x)=y]:
f(-5)=100 f(-3)=-10 f(-2)=-2 f(2)=10 f(4)=-90 f(7)=-10
Jetzt brauchen Sie eine Funktion, dessen Graph durch die festgelegten Punkte führt. Sie kennen die Normalform für ganzrationale Funktionen 5. Grades.

Sie lautet:
f(x)=a*x 5*x 4c*x 3d*x 2+e*x+f
Das kann auch so geschrieben werden:
x 5*a+x 4*b+x 3*c+x 2*d+x*e+f=y
Sie haben sechs verschiedene X-Werte und die daraus resultierenden Y-Werte. Sie setzen nun sechsmal die X-Werte in jeweils einer neuen Gleichung ein und setzen diese gleich dem dazugehörigen Y-Wert.

Das sieht dann so aus:
1. Gleichung:
-3125a+625b-125c+25d-5e+f=100
2. Gleichung:
-243a+81b-27c+9d-3e+f=-10
3. Gleichung: -32a+16b-8c+4d-2e+f=-2
4. Gleichung: 32a+16b+8c+4d+2e+f=10
5. Gleichung:
1024a+256b+64c+16d+4e+f=-90
6. Gleichung:
16807a+2401b+343c+49d+7e+f=-10

Um nun a, b, c, d, e und f zu ermitteln, müssen alle Gleichungen nach dem Gauschen Eliminationsverfahren aufgelöst werden. Diese zeitaufwendige Aufgabe übernimmt der Computer.
Er fragt nach der Anzahl der Unbekannten. Sie müssen hier eine >6< eintippen. Sie geben danach die Koeffizienten der Gleichungen ein. Es wird eine Gleichung nach der anderen eingegeben. Der Rechner fragt immer zuerst nach den Koeffizienten der jeweiligen Gleichung [Gl. 1/Koef. 3=].
Dann fragt er nach dem f(x)-Wert der Gleichung [Gl. 1/f(x) =].
Geben Sie bei jeder Frage nur einen Wert ein!

Die Koeffizientenmatrix:

	Koef. 1	Koef. 2	Koef. 3	Koef. 4	Koef. 5	Koef. 6		f(x)
Gleichung 1	-3125	625	-125	25	-5	1	:	100
Gleichung 2	-243	81	-27	9	-3	1	:	-10
Gleichung 3	-32	16	-8	4	-2	1	:	-2
Gleichung 4	32	16	8	4	2	1	:	10
Gleichung 5	1024	256	64	16	4	1	:	-90
Gleichung 6	16807	2401	343	49	7	1	:	-10

Nach 40 Minuten Rechenzeit wird folgendes Ergebnis angezeigt:

a = .012962963
b = .2767195767
c = -1.616137566
d = -8.94021164
e = 9.257142857
f = 35.33333333

Somit lautet die Funktionsgleichung:

f(x) = .012962963*x^5+.2767195767*x^4-1.616137566*x^3
-8.94021164*x^2+9.257142857*x+35.33333333

Falls Sie sich das Ergebnis nicht ausdrucken lassen, müssen Sie sich die Werte unbedingt notieren!
Der Rechner führt Sie jetzt mittels verschiedener Menues durch das Programm:

Beschreibung des Hauptmenues

Nach dem Programmstart erscheint ein 5 Punkte umfassendes Menue auf dem Monitor.

1. Unter Punkt eins geben Sie dem Rechner Ihre Funktionsgleichung ein. Wenn Sie Taste >1< drücken, gelangen Sie in ein Untermenue, drücken Sie hier am besten die >5<.
2. Unter Punkt zwei weist der Rechner darauf hin, daß er ausschließlich für ganzrationale Funktionen programmiert ist.
   Falls Sie mit anderen Funktionstypen arbeiten wollen, müssen Sie folgendes ändern:
   Die Zeilen 550→560:
   Hier werden die Stammfunktion und die drei Ableitungen definiert.
   Y(X)=Stammfunktion
   AB1=1. Ableitung
   AB2=2. Ableitung
   AB3=3. Ableitung
   Die Zeilen 610→660:
   Hier wird aus den Funktionskoeffizienten ein String gebildet, um die Funktion jederzeit vor Augen zu haben. Haben Sie einen anderen Funktionstyp definiert, so müssen Sie die Stringausdrücke entsprechend umändern.
3. Punkt drei sollten Sie immer dann drücken, wenn Sie die Funktion schon eingegeben haben und mit ihr arbeiten wollen. Sie gelangen in ein Untermenue. (Beschreibung folgt!)
4. Wenn Sie alle berechneten Werte auf dem Bildschirm sehen wollen, so drücken Sie Taste >4<. Ein weiteres Menue sagt, was Sie zu tun haben.
5. Falls Sie die Funktion noch nicht kennen und nur einige Punkte zur Verfügung haben, können Sie diesen Punkt wählen.
   Das angewählte Unterprogramm berechnet die Koeffizienten Ihrer Funktion. (siehe Beschreibung!)
   Mit diesem Unterprogramm können Sie zudem lineare Gleichungssysteme mit bis zu 10 Unbekannten berechnen.

Beschreibung des Untermenues:

Die Unterprogrammpunkte 0→8 sind eindeutig. Falls notwendig beschreibt sich das Programm von selbst.

6. Haben Sie Punkt >6< gewählt, so erscheint eine Eingabemaske. Ist die hier definierte Grafik erstellt und Sie wollen mit anderen Achsenabschnitten arbeiten, so drücken Sie die Gleichheitstaste. Sie gelangen dann wieder zur Eingabemaske.
7. Unter Punkt sieben gelangen Sie mit der '='-Taste aus der Tabelle.
8. Nachdem die Funktion vollständig diskutiert ist können Sie die Kurvendiskussion ausdrucken lassen.

```
1
100 OPTION BASE 1 :: DIM NL(5),MA(5),MI(5),WN(5),U(10,10),BF(10),CF(10),AL(10)::
 OPEN #1:"RS232.DA=8" :: ON WARNING NEXT :: ON ERROR 1690
110 CALL CLEAR :: CALL SCREEN(16):: FOR I=0 TO 12 :: CALL COLOR(I,16,5):: NEXT I
 :: CALL COLOR(13,10,5,14,2,5):: GOSUB 1390 :: DEF @$(I)=CHR$(I):: DEF S$(I)=STR
$(I)
120 SY=0 :: DISPLAY AT(1,1)ERASE ALL BEEP:" komplette Funktionsanalyse  "&RPT$(@
$(127),4)&@$(129)&"(bis zum 5.Grad)"&@$(128)&RPT$(@$(127),4):: CALL HCHAR(3,9,13
0,16)
130 CALL VCHAR(1,32,127,48):: DISPLAY AT(6,1):"1] Eingabe der Koeffizienten" ::
CALL HCHAR(7,6,127,25)
140 DISPLAY AT(10,1):"2] Funktion anderen Typs" :: CALL HCHAR(11,6,127,21):: DIS
PLAY AT(14,1):"3] "&@$(129)&"Funktion beibehalten"&@$(128)
150 CALL HCHAR(15,6,127,22)
160 DISPLAY AT(18,1):"4] Wiederholung der Werte" :: CALL HCHAR(19,6,127,22):: DI
SPLAY AT(22,1):"5] lineares Gleichungssystem" :: CALL HCHAR(23,6,127,25)
170 CALL KEY(0,K,S):: IF S=0 THEN 170 ELSE IF K=51 THEN 500 ELSE IF K=49 THEN SY
=1 :: GOTO 500 ELSE IF K=52 THEN 1250 ELSE IF K=53 THEN 180 ELSE IF K=50 THEN 48
0 ELSE 170
180 DISPLAY AT(1,1)ERASE ALL BEEP:" lineares Gleichungssystem:  "&RPT$(@$(127),2
6): : : :"Analyse bis zum zehnten Grad"&RPT$(@$(127),28)
190 DISPLAY AT(12,1):" Moechten Sie einen Drucker  benutzen ? (J/N)   "&RPT$(@$(
127),7)
200 CALL KEY(0,K,S):: IF S=0 THEN 200 ELSE IF K=74 THEN DR=1 ELSE IF K=78 THEN D
R=0 ELSE 200
210 DISPLAY AT(18,1):"Anzahl der Unbekannten :" :: ACCEPT AT(18,26)BEEP SIZE(-2)
:W :: IF W<2 OR W>10 THEN 210
220 CALL CLEAR :: PRINT :"Eingabe des Gleichungs-":"systems:":
230 FOR P=1 TO W :: FOR S=1 TO W :: INPUT "Gl."&S$(P)&"/Koef."&S$(S)&"=":U(P,S):
: NEXT S :: INPUT "Gl."&S$(P)&"/f(x)=":AL(P):: PRINT :: NEXT P :: CALL CLEAR
240 A$="Gleichungssystem mit "&S$(W)&" Unbekannten:"
250 PRINT SEG$(A$,1,20):SEG$(A$,22,LEN(A$)-21): : :: IF DR=1 THEN PRINT #1: : :A
$:RPT$("=",LEN(A$)): :
260 FOR P=1 TO W :: IF DR=1 THEN PRINT #1: :"Koeffizienten der "&S$(P)&". Gleich
ung:":TAB(5);
270 FOR S=1 TO W :: PRINT INT(U(P,S));:: IF DR=1 THEN PRINT #1:U(P,S);
280 NEXT S :: PRINT "|";AL(P):: IF DR=1 THEN PRINT #1:"|";AL(P)
290 NEXT P :: PRINT :: IF DR=1 THEN PRINT #1
300 FOR N=W TO 2 STEP -1 :: GOSUB 380 :: IF DE=0 THEN 350
310 T=DE :: FOR P=1 TO N :: CF(P)=U(P,N):: U(P,N)=AL(P):: NEXT P :: GOSUB 380 ::
 V(N)=DE/T :: FOR P=1 TO N-1 :: AL(P)=AL(P)-V(N)*CF(P):: NEXT P :: NEXT N
320 V(1)=AL(1)/U(1,1):: A$="Die Loesung der Gleichung " :: B$="lautet:" :: PRINT
 A$:B$: : :: IF DR=1 THEN PRINT #1: :A$;B$: :
330 FOR I=1 TO W :: PRINT @$(96+I);"=";V(I):: IF DR=1 THEN PRINT #1:TAB(6);@$(96
+I);"=";V(I)
340 V(I)=0 :: NEXT I :: GOTO 360
350 A$="Es gibt keine eindeutige" :: B$=" Loesung der Gleichung." :: PRINT "  ";
A$:"  ";B$: : : : : : :: IF DR=1 THEN PRINT #1:A$;B$
360 PRINT :: DISPLAY AT(24,1):"neue Gleichung ? (J/N) :"
370 ACCEPT AT(24,26)SIZE(-1):A$ :: IF A$="J" THEN 180 ELSE IF A$="N" THEN CALL C
LEAR :: GOTO 120 ELSE 370
380 DE=0 :: CE=1 :: FOR I=1 TO N :: CE=CE*I :: BF(I)=I :: NEXT I
390 FOR R=1 TO CE STEP 2 :: O=N-2 :: K=0 :: FOR I=1 TO N-1 :: FOR Q=I+1 TO N ::
IF BF(I)<BF(Q)THEN 400 ELSE K=K+1
400 NEXT Q :: NEXT I :: GOSUB 470 :: H=BF(N):: BF(N)=BF(N-1):: BF(N-1)=H :: K=K+
1 :: GOSUB 470
410 IF O=0 THEN RETURN ELSE BF(O)=BF(O)+1
420 FOR I=1 TO O-1 :: IF BF(I)=BF(O)THEN 410
430 NEXT I :: IF BF(O)<=N THEN 440 ELSE O=O-1 :: GOTO 410
440 H=O :: FOR I=1 TO N :: FOR Q=1 TO O :: IF BF(Q)=I THEN 460
450 NEXT Q :: H=H+1 :: BF(H)=I
460 NEXT I :: NEXT R :: RETURN
470 L=1 :: FOR I=1 TO N :: L=L*U(I,BF(I)):: NEXT I :: DE=DE+(-1)^K*L :: RETURN
480 GOSUB 1350 :: DISPLAY AT(5,1):"Sie muessten dann Statementsin den Zeilen: 55
0-560 sowiein den Zeilen: 610-660 etwasveraendern."
```

```
490 DISPLAY AT(12,1):"In der Dokumentation finden Sie Informationen darueber." :
: GOTO 1220
500 GOSUB 1350 :: DISPLAY AT(4,2):"Menue:":" "&RPT$(@$(127),6)&RPT$(" ",14)&"0] 
Exit"
510 DISPLAY AT(7,1):"1] Y-Achsenabschnitt": :"2] Nullstellen": :"3] Extremstelle
n": :"4] Wendestellen"
520 DISPLAY AT(15,1):"5] 9esamte Kurvendiskussion": :"6] 9raPhische Darstellun9"
: :"7] Wertetabelle": :"8] Flaechenberechnun9": :"9] ComPuterbericht.....>5< ?"
530 CALL KEY(0,T,S):: IF S=0 OR T<48 OR T>57 THEN 530
540 CALL CLEAR :: IF SY=0 THEN 670
550 DEF Y(X)=A*X^5+B*X^4+C*X^3+D*X^2+E*X+F :: DEF AB1=5*A*X^4+4*B*X^3+3*C*X^2+2*
D*X+E :: DEF AB2=20*A*X^3+12*B*X^2+6*C*X+2*D
560 DEF AB3=60*A*X^2+24*B*X+6*C :: F$="="
570 DISPLAY AT(2,1):"Ein9abe der Koeffizienten  einer 9anzrationalen         Fun
ktion mit der Form:" :: CALL HCHAR(1,3,127,25):: CALL HCHAR(5,3,127,25)
580 DISPLAY AT(7,1):"f(x)=a*x^5+b*x^4+c*x^3+d*x^2+e*x+f":"":"":"a =":"":"b =":""
:"c =":"":"d =":"":"e =":"":"f ="
590 ACCEPT AT(11,5)BEEP:A :: ACCEPT AT(13,5)BEEP:B :: ACCEPT AT(15,5)BEEP:C :: A
CCEPT AT(17,5)BEEP:D :: ACCEPT AT(19,5)BEEP:E :: ACCEPT AT(21,5)BEEP:F
600 CALL CLEAR :: MX,MN,NU,WE=0
610 IF A<>0 THEN F$=F$&S$(A)&"*x^5"
620 IF B>0 THEN F$=F$&"+"&S$(B)&"*x^4" ELSE IF B<>0 THEN F$=F$&S$(B)&"*x^4"
630 IF C>0 THEN F$=F$&"+"&S$(C)&"*x^3" ELSE IF C<>0 THEN F$=F$&S$(C)&"*x^3"
640 IF D>0 THEN F$=F$&"+"&S$(D)&"*x^2" ELSE IF D<>0 THEN F$=F$&S$(D)&"*x^2"
650 IF E>0 THEN F$=F$&"+"&S$(E)&"*x" ELSE IF E<>0 THEN F$=F$&S$(E)&"*x"
660 IF F>0 THEN F$=F$&"+"&S$(F)ELSE IF F<>0 THEN F$=F$&S$(F)
670 PRINT "Funktion: f(x)"&F$: :
680 ON T-47 GOTO 120,690,700,810,960,690,1090,1360,1060,1520
690 PRINT "Y-Achsenabschnitt:":"Py=(0|"&S$(Y(0))&")" :: PRINT :: IF T<>53 THEN 1
220
700 GOSUB 1240 :: GOTO 720
710 S=S/10 :: V1=V2 :: V3=V4
720 FOR X=V1 TO V3 STEP S :: V4=X
730 IF Y(X)=0 OR S=1E-9 THEN 750 ELSE IF SGN(V5)=0 THEN 740 ELSE IF SGN(Y(X))<>S
GN(V5)THEN 710
740 V5=Y(X):: V2=V4 :: NEXT X :: GOTO 770
750 PRINT "Nullstelle bei x=";X :: NU=NU+1 :: NL(NU)=X :: CALL SOUND(50,420,0)
760 X=X+.5 :: V1=X :: V3=20 :: K=K+1 :: V5=0 :: S=1 :: GOTO 720
770 IF K<>0 THEN 790
780 PRINT "keine Nullstellen!"
790 IF T<>53 THEN 1220
800 PRINT
810 GOSUB 1240 :: GOTO 830
820 S=S/10 :: V1=V2 :: V3=V4
830 FOR X=V1 TO V3 STEP S :: V4=X
840 IF AB1=0 OR S=1E-8 THEN 860 ELSE IF SGN(V6)=0 THEN 850 ELSE IF SGN(AB1)<>SGN
(V6)THEN 820
850 V6=AB1 :: V2=V4 :: NEXT X :: GOTO 920
860 IF AB2>=0 THEN 880
870 PRINT "relatives Maximum bei":"("&S$(X)&"|"&S$(Y(X))&")" :: MX=MX+1 :: MA(MX
)=X :: GOTO 900
880 IF AB2=0 THEN 910
890 PRINT "relatives Minimum bei":"("&S$(X)&"|"&S$(Y(X))&")" :: MN=MN+1 :: MI(MN
)=X
900 CALL SOUND(50,420,0)
910 K=K+1 :: X=X+.1 :: V1=X :: V3=20 :: V6=0 :: S=1 :: GOTO 830
920 IF K<>0 THEN 940
930 PRINT "keine Extremstellen!"
940 IF T<>53 THEN 1220
950 PRINT
960 GOSUB 1240 :: GOTO 980
970 S=S/10 :: V1=V2 :: V3=V4
980 FOR X=V1 TO V3 STEP S :: V4=X
990 IF AB2=0 OR S=1E-9 THEN 1010 ELSE IF SGN(V7)=0 THEN 1000 ELSE IF SGN(AB2)<>S
GN(V7)THEN 970
```

```
1000 V7=AB2 :: V2=V4 :: NEXT X :: GOTO 1040
1010 IF AB3=0 THEN 1030
1020 PRINT "Wendestelle bei":"("&S$(X)&"|"&S$(Y(X))&")" :: WE=WE+1 :: WN(WE)=X :
: CALL SOUND(50,420,0)
1030 K=K+1 :: X=X+.1 :: V1=X :: V3=20 :: V7=0 :: S=1 :: GOTO 980
1040 IF WE=0 THEN PRINT "keine Wendestellen!": :
1050 GOTO 1220
1060 PRINT "Flaechenberechnung:": : :: FL,V8,V9=0 :: INPUT "untere Grenze: Xu=":
G1 :: INPUT "obere  Grenze: Xo=":G2 :: QQ=Y(G1)+Y(G2)
1070 FOR Q=G1+.02 TO G2-.02 STEP .02 :: V8=V8+Y(Q):: NEXT Q :: V8=V8*2
1080 FOR R=G1+.01 TO G2-.01 STEP .02 :: V9=V9+Y(R):: NEXT R :: V9=V9*4 :: FL=(QQ
+V9+V8)/300 :: PRINT :"Flaeche unter dem Graphen:  A =";FL:"" :: GOTO 1220
1090 DISPLAY AT(1,1)ERASE ALL:"graphische Darstellung:":"":"":"f(x)"&F$ :: CALL
HCHAR(2,3,127,23)
1100 DISPLAY AT(9,1):"untere Grenze (X-Achse)      "&RPT$(@$(127),23):"Xu =":"":"
obere  Grenze (X-Achse)      "&RPT$(@$(127),23):"Xo ="
1110 DISPLAY AT(17,1):"untere Grenze (Y-Achse)      "&RPT$(@$(127),23):"Yu =":"":
"obere  Grenze (Y-Achse)      "&RPT$(@$(127),23):"Yo ="
1120 ACCEPT AT(11,6)BEEP:A1 :: ACCEPT AT(15,6)BEEP:B1 :: ACCEPT AT(19,6)BEEP:C1
:: ACCEPT AT(23,6)BEEP:D1 :: CALL CLEAR
1130 P1=INT((24/(ABS(D1)+ABS(C1)))*ABS(D1))
1140 P2=(32/(ABS(B1)+ABS(A1)))*ABS(A1)+1 :: P3=0
1150 IF P1<1 OR P1>24 OR P2<1 OR P2>32 THEN 1090
1160 DISPLAY AT(24,1):"$="&S$(INT(((B1-A1)/30)*100+.5)/100)&" #="&S$(INT(((D1-C1
)/23)*100+.5)/100)
1170 CALL HCHAR(P1,1,36,31):: CALL VCHAR(1,P2,35,23):: CALL HCHAR(P1,P2,37):: CA
LL HCHAR(P1,31,39):: CALL HCHAR(1,P2,38)
1180 FOR I=A1 TO B1 STEP (ABS(B1)+ABS(A1))/32
1190 P4=P1-INT((24/(ABS(C1)+ABS(D1)))*Y(I)):: P3=P3+1 :: IF P3>32 THEN 1210 ELSE
 IF P4<=0 OR P4>24 THEN 1200 ELSE CALL HCHAR(P4,P3,143)
1200 NEXT I
1210 CALL KEY(0,K,S):: IF S=0 THEN 1210 ELSE IF K=61 THEN 1090
1220 DISPLAY AT(24,1)SIZE(13):"Menue?  (J/N)"
1230 CALL KEY(0,K,S):: IF S=0 THEN 1230 ELSE IF K=74 THEN CALL CLEAR :: GOTO 120
 ELSE IF K=78 THEN CLOSE #1 :: END ELSE 1230
1240 S=1 :: V5,V6,V7,K=0 :: V1=-26.12 :: V3=26 :: RETURN
1250 DISPLAY AT(1,7)ERASE ALL:"Kurvendiskussion:" :: CALL HCHAR(3,1,61,32):: DIS
PLAY AT(6,1):"1] Nullstellen":RPT$(@$(127),14)
1260 DISPLAY AT(9,1):"2] Extremstellen":RPT$(@$(127),16)
1270 DISPLAY AT(12,1):"3] Wendestellen":RPT$(@$(127),15): :"4] Flaeche unter dem
 Graphen"&RPT$(@$(127),28)
1280 CALL KEY(0,K,S):: IF S=0 OR K<49 OR K>52 THEN 1280
1290 GOSUB 1350 :: DISPLAY AT(5,1):"Funktion: f(x)"&F$ :: ON K-48 GOTO 1300,1310
,1330,1340
1300 FOR I=1 TO NU :: DISPLAY AT(I+9,1):"Nullstelle x=";NL(I):: NEXT I :: GOTO 1
220
1310 DISPLAY AT(10,1):"rel. Maximum:" :: FOR I=1 TO MX :: DISPLAY AT(I+10,1):"("
:MA(I);"|";Y(MA(I));")" :: NEXT I
1320 DISPLAY AT(11+MX,1):"rel. Minimum:" :: FOR I=1 TO MN :: DISPLAY AT(I+11+MX,
1):"(";MI(I);"|";Y(MI(I));")" :: NEXT I :: GOTO 1220
1330 DISPLAY AT(10,1):"Wendestelle:" :: FOR I=1 TO WE :: DISPLAY AT(I+10,1):"(";
WN(I);"|";Y(WN(I));")" :: NEXT I :: GOTO 1220
1340 DISPLAY AT(10,1):"Abzissen-,X-Achsenintervall:":"[";S$(G1)&"|";S$(G2)&"]":
:"Flaeche =";FL :: GOTO 1220
1350 DISPLAY AT(1,7)BEEP ERASE ALL:"Kurvendiskussion:" :: CALL HCHAR(3,3,61,28):
: RETURN
1360 GOSUB 1350 :: DISPLAY AT(5,3):"Wertetabelle:": :" X-Werte:","Y-Werte:" :: C
ALL HCHAR(8,3,130,28):: CALL VCHAR(9,16,131,15):: QQ=0
1370 QQ=QQ+1 :: ACCEPT AT(QQ+9,2)BEEP VALIDATE(NUMERIC,"=")SIZE(10):A$
1380 IF QQ>13 OR A$="=" THEN 1220 ELSE DISPLAY AT(QQ+9,17):Y(VAL(A$)):: GOTO 137
0
1390 CALL CHAR(35,"1010107010101010",36,"000000FF101",37,"101010FF1010101",38,"00
10387CFE10101",39,"10181CFE1C181",143,"00183C7E7E3C18")
1400 CALL CHAR(96,"102844007C007C",59,"0010101010101010",127,"000000FF",128,"1030
70FF70301",129,"080C0EFF0E0C08",130,"0000000000000000FF")
```

```
1410 CALL CHAR(131,"101010101010101")
1420 FOR I=65 TO 90 :: READ A$ :: CALL CHAR(I,A$):: NEXT I :: FOR I=97 TO 122 ::
 READ A$ :: CALL CHAR(I,A$):: NEXT I :: FOR I=48 TO 57 :: READ A$ :: CALL CHAR(I
,A$):: NEXT I :: RETURN
1430 DATA 003844447C4444,007844784444 78,00384440404438,00782424242478,007C407840
407C,007C407840404
1440 DATA 003844405C4438,0044447C444444,007C101010107C,007C0404044438,0048506050
4844,0040404040407C
1450 DATA 00446C54444444,004464544C4444,00384444444438,007844447840 4,00384444544
C38,007844447848 44,00384038044438
1460 DATA 007C101010101,0044444444438,00424242422418,00424242425A24,00422418182
442,00412214080808,007E040810207E
1470 DATA 000038043C443E,004040784444 78,00003840404038,0004043C44443C,0000384478
403C,001C207820202,00003C44443C0438
1480 DATA 004040784444 44,00100030101038,00080018080828 10,00202428302824,00202020
20201C,00006854545454,00007844444444
1490 DATA 000038444444438,000078444478404,00003C44443C0404,00002C3020202,00003840
380478,00207820202018,00004444444438,0000444444281
1500 DATA 000044445454 28,00004428102844,00004444 4C340438,00007C0810207C,00384C54
546438,00081808080 81C,00384404384 07C
1510 DATA 00384418044438,00081828487C08,007C4078044438,00384078444438,007C040810
2020,00384438444438,003844443C0438
1520 PRINT :: DISPLAY AT(1,7):"ComPuterbericht:":"       "&RPT$(@$(127),16):;:"Dr
uckersystem initialisierenund >ENTER< druecken!"
1530 CALL KEY(0,K,S):: IF K<>13 THEN 1530
1540 PRINT #1:@$(16);"15";@$(14);"Kurvendiskussion";@$(15);"(ComPuterbericht):":
@$(16);"15";@$(14);RPT$("=",25);@$(15):""
1550 PRINT #1:"Die Funktion hat die Gleichung:" :: PRINT #1:"f(x)"&F$: :
1560 PRINT #1:"":@$(16);"22";@$(14);"Y-Achsenabschnitt:":@$(16);"22";"----------
---------";@$(15):""
1570 PRINT #1:"Der Y-Achsenabschnitt lautet: f(0)=";Y(0):""
1580 PRINT #1:"":@$(16);"28";@$(14);"Nullstellen:":@$(16);"28";"----------";@$
(15):"" :: IF NU=0 THEN 1610
1590 PRINT #1:"":"Nullstelle bei:"
1600 FOR I=1 TO NU :: PRINT #1:@$(16);"17";"x =";NL(I):: NEXT I :: GOTO 1620
1610 PRINT #1:"Es gibt keine Nullstellen!"
1620 PRINT #1:"":"":@$(16);"26";@$(14);"Extremstellen:":@$(16);"26";"----------
---";@$(15):: IF MX=0 THEN PRINT #1:"":"Es gibt kein rel. Maximum!" :: GOTO 1640
1630 PRINT #1:"":"rel. Maximum bei:" :: FOR I=1 TO MX :: PRINT #1:@$(16);"19";"(
";MA(I);"|";Y(MA(I));")" :: NEXT I
1640 IF MN=0 THEN PRINT #1:"":"Es gibt kein rel. Minimum!" :: GOTO 1660
1650 PRINT #1:"":"rel. Minimum bei:" :: FOR I=1 TO MN :: PRINT #1:@$(16);"19";"(
";MI(I);"|";Y(MI(I));")" :: NEXT I
1660 PRINT #1:"":"":@$(16);"27";@$(14);"Wendestellen:":@$(16);"27";"----------
-";@$(15):"" :: IF WE=0 THEN PRINT #1:"Es gibt keine Wendestelle!" :: GOTO 1680
1670 PRINT #1:"Wendestelle bei:" :: FOR I=1 TO WE :: PRINT #1:@$(16);"18";"(";WN
(I);"|";Y(WN(I));")" :: NEXT I
1680 PRINT #1: : :@$(16);"10";RPT$("-",60): : : :: GOTO 1220
1690 CALL SOUND(400,110,0,220,0,330,0):: ON ERROR 1690 :: RETURN NEXT
```

Jetzt überall im Zeitschriften-handel:

Das Softwaremagazin

Hallo, liebe Rätselfreunde ...

Wie in jeder Ausgabe von CPU hat die Redaktion auch diesmal wieder ein recht kniffeliges Rätsel für Euch zusammengestellt. Dieses Suchrätsel wurde von einem unserer Computer so zusammengestellt, daß die zu findenden Begriffe aus der Computerwelt in waagerechter, senkrechter als auch in diagonaler Stellung auffindbar sind. Um dieses Rätsel nicht allzu schwer zu gestalten, haben wir Euch die versteckten Begriffe unten angegeben. Eure Aufgabe ist es nun, die Positionen dieser Begriffe herauszufinden und uns diese Seite, bzw. eine Kopie der Seite, zuzusenden.

Folgende Begriffe sind zu finden:

BASIC COMPUTER MODEM SOFTWARE APPLE COMMODORE ATARI ASSEMBLER MONITOR DISKETTE PRINT CPU-HAPPY KASSETTE HARDWARE MODUL AKKUSTIKKOPPLER FLOPPY HACKER TASTATUR DRUCKER

und wie gesagt: horizontal, vertikal, diagonal

Schreiben Sie auch diesmal wieder die Lösung auf eine Postkarte, frankieren sie ausreichend und schicken sie an den Tronic-Verlag, Stichwort CPU Happy 3285, Postfach 41, D-3444 Wehretal 1.

Alle richtigen Lösungen nehmen an der Verlosung der Programmkassetten teil. Mitmachen darf jeder, mit Ausnahme der Mitarbeiter des Tronic-Verlages sowie deren Angehörige. Einsendeschluß ist der 18. 3. 1985.

Der Rechtsweg bleibt wie üblich ausgeschlossen.

1. Preis
10 Kassetten aus unserem Software-Service

2. Preis
3 Kassetten aus unserem Software-Service

3. Preis
2 Kassetten aus unserem Software-Service

4.-30. Preis
1 Kassette aus unserem Software-Service

```
ZQMGYVGSNFRUKHVCXFXNQKALIAPGOFMFSEBWRKZRTXULJGEMQU
YPOIPBKSNPWJUIBGMMFVVDXCYUYKOPGKGDUMGXEMMJSIHUTBEF
OEAJSLXKQUTOPYFTLJHHTMCFUYHRRICNBEIFHAPACSTZGXRPIP
ULFPPQZZSAKTKGIGFUITHWIHTAQIIMSYPQCERKBSRWTUUOLDRC
UIHEZIIAYVMDUQPVCTDFQZRGIHNKQQBMDYIUSOZTWWSJYQZHWM
QBOYVBAWCCYTNKTITEJTYFVUATFYYZDZIHYIMMWUPFFSCOOJAK
VARQRPIEFNUSXTQBHEBVBFMCGBFJXVIKXUSQJWSVDOWEVMGPAA
UABXQGWCAAGVHSVWPGKXBKKMSDDFDWDNVMRGZXXSBLJCJPDDNX
XVXIJDDTWELTFMEDJXHUTELFOWOFBFVEKMKDTPGJBLCWCJIKKI
LEBHWGWPNXGKUMJEGIGNRVFNZEWKBTXXCLMDEHUHANWNEFPXRW
LWGPMIWTKWSLGHZKPRXYXTRJHGYKXPOTREMPHJVMKHBQBTSUAY
BYPAZNWTACHGDMENHJCJWZGZOQFTIALFCVHNTCINBSHZIQOFZC
YYHLOVRESORTUFIWBFLQKDFVVKAJORZBSAAAAEBALABRGIYSPK
AHVSTOHISGMVZFHUZWZTMPSHMUVBGSRARSNAEFXLSITQBSLMZX
SKXTCGQVEASIYYDQSANOWODHAUKKMGXYSCOLLNMSQDZNZSCMPZ
BMLVLXDOTLHSRYIZBQDATYVMTROTAENDTQLYSDEBWTXWUGEIWM
QLBJJYMMTGFJAVPBWEUZASIAWERBUNYMFVJSOMNVLSTPWYIOLH
INIXUNHHEZJNSMAPMMMGPKKTYSRHPCVTFVJMBOSUJWTJMSMKFL
BPELOWSOFTWAREVSIJWXTUDMTPHOJVZEXUQLNHMAQDXUEEXEDR
VPGDXSQZCTYXRLZHHLEXOTYARZIHUTALYSERTWXOSTSLQTOZUE
VUEXVPDIRHQYNTRGGHLGPLTLSUNKCXJRORFUONONTUGVJHZLUI
GEZYSSUGDLUCLQDRUCKERUPHCKMTCBHVQODJUJOTFECTQJIXXD
OZPEOIYPIVTGXZIKIQLHRWYJKMUSGWRETITWARUAEIICCBABKU
JKWHCVNUSSCCOAUYYQWPHLOKGSKIOYCOKBILJXVYHKCZQNFVMI
DEYGFKREKWABVWKWIQPVVDGAQDGEHJHERQLAAGAQSVOAMROAXU
BBGBDXLJEPSOMGSCORZCPOPPTHXHDSQEBHSVIKCVRMIEUKATFG
REUPISRJTCCRDEVARWTWBGVVGAWGJQYSNUJOZQKKWGOYTTYJAE
XWKMQRNATDBRRVBKFJVHXFVECECRPBLYAXTXZLVUAZUGSHHXSG
ZGEGTKYWEXUHLFOYPCXPFCQYKYRTQMRBQXLVALNHSYMOFDLYXJ
GJLXMLAPWWGBHKGYDLKBSURMSILGPIYHOUOFOWVIDTKYSDZZHF
WNJFMIXPDUSTARCOTIKXAYTYYFRPJITYQZOITLGNXRIAJODIPX
VTJLJOWNLWHBVGXTKDFPFCAHYCVVUYJHKHHWSADCTXIKNBFHXD
HXHVKCOEBWLCZJISHLOIORUSBPWPMJJFOEZIPNMVEUDCKIKQYF
OULORCBXJDMZYZMIOXUEEEHHIUSMZTWVJUHBMNTCLPWPZOEJMW
JJHWQSOSCFGWYSMPODKGRGUYW-OJXFVFEWAVMBNKMEFMLXPMSK
IWPPKKOAIORFHHPKYSPIKBTYKHNXGPZQVTOUTAOERDHAKEOLMR
AMMRJJKCSLANNYEPNAMODULZFALGOARUBEHIJTQLNVWIEFVUEJ
PYMIMPJFBBKHRTDOIGMYOHKNOPSNDSYAPEMBRAQEWLPZOOKRNR
WZEPVBYNJIEPMXWDVQAOPALRYPVBLCYBGHLWKRTVBHDYDKZIIX
CLAXGJXHQGAKZOTEVGOOHRNUEYFTADKQVQLBKIYYGMVFIAUJLI
YFRWDWXZLSNLAHNZYDATODSTGDVVNSPSXYSTBIGNZWTSNIPEVK
LSHNIATYWOONVPRIKVLAWWQJCKBAEWIILOJXMOWCENOBNXAZXJ
JOJXPPHWCSWDJLYNTDKYIAJUDQYPBPDCPYDVXZYJSECPSLUNCI
FKGPGEQIUTPDOOSKFOJWYRQFKXBHMNUFPUDMPAQRQJAOLKAARP
HFLRAIPQAKQAOETIEQRPXEBYCBCRHIOYGVKIFQBVSICTMSWKDU
WEOJEKVEJFKDUJHPVDKWMXQRCFDHDLXXUGZTUTBSGEDIPPGIQY
MUDLGTELSLOFIOGSWGBECDSMLEKWTTNJQWNJHRNNERJNQRUIEV
TWFBYQDBPOBDAZKVNWQIGDBTGYORUVVHDTFBXKVUIBJXUEDTOS
VOSUXKZDLZKWJOLZIYMCVCDTOKMNHIPQVFQBDOFQAWYBYGVQED
IJGHQBEHCAGSHLNXOULYTDYYZLRGHBZFZQREPMLCOMMODORETR
```

Kassettenservice

Tronic-Verlag
Landstraße 29
3444 Wehretal 1

Unser Paketservice
Kaum zu glauben

Für unsere Leser, die Überraschungen lieben, hat sich unsere Redaktion etwas einfallen lassen. Überzeugen Sie sich selbst. Wir haben Ihnen einige S u p e r-Angebote zusammengestellt:

Spitzen-Programme zum Spitzen-Preis:

Bestell-Nr. 100	**kleines Programm-Paket**	3 bespielte Kassetten DM 27,50 3 bespielte Disketten DM 48,–
Bestell-Nr. 110	**großes Programm-Paket**	8 bespielte Kassetten DM 64,50 8 bespielte Disketten DM 99,–
Bestell-Nr. 200	**exklusive Disketten-Box**	inkl. 8 bespielte Disketten zum Preis von DM 148,–
Bestell-Nr. 210	**exklusive Disketten-Box**	inkl. 10 Leerdisketten zum Preis von DM 99,–

Dieses Angebot gilt nur für Commodore 64, Atari, TI-99 und Apple!

Bestell-Nr. 300 **C o m p u t r o n i c-Angebot** unsere Ausgaben Heft 4–10 zum Sparpreis von DM 26,–!

Computer-Messen '85

25. 2.– 1. 3. 85	didacta 85/Stuttgart
20. 3.–24. 3. 85	Hobby-Tronic/Computerschau/Dortmund
17. 4.–24. 4. 85	Hannovermesse
22. 5.–24. 5. 85	komm 85/Berlin
13. 6.–16. 6. 85	Internationale Computer-Show/Köln
30. 8.– 8. 9. 85	Internationale Funkausstellung Berlin
24. 9.–27. 9. 85	CAMP/Computer Graphics/Berlin
9. 10.–13. 10. 85	Hobby-Elektronik 85/ Stuttgart
9. 10.–14. 10. 85	Frankfurter Buchmesse
17. 10.–20. 10. 85	CFA – Commodore Fachausstellung/Frankfurt
28. 10.– 1. 11. 85	SYSTEMS 85/München

Computermessen im Ausland

1. 3.–16. 3. 85	4th Australian Personal Computer Show/Sidney
26. 3.–28. 3. 85	COMDEX Japan/Tokio
31. 3.– 3. 4. 85	SOFTCON/Atlanta, USA
1. 6.– 4. 6. 85	CES – Consumer Electronic-Show/Chicago, USA
Juni 85	PET World Show/London
September 85	SICOB/Paris
18. 9.–22. 9. 85	PCW/London
Oktober 85	COMDEX/Las Vegas, USA

Steuertips für Computerbenutzer

Personal-, Mikro- und Homecomputer halten in immer größerem Maße Einzug in Büros und werfen somit natürlich auch steuerrechtliche Probleme auf?

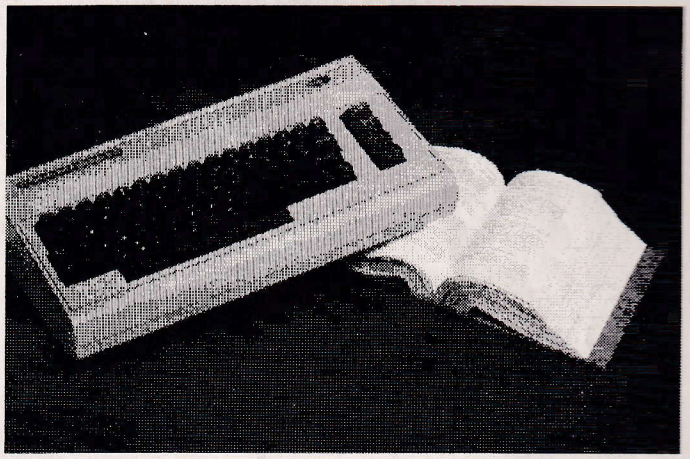
Während die steuerliche Behandlung von Computern (sprich der Hardware) bei Selbständigen eindeutig geregelt ist, gibt es bei der Abschreibung der benötigten Software noch einige Probleme.

Der Kauf von neuen oder gebrauchten Computern ist bei überwiegend betrieblicher Nutzung in Höhe des Kaufpreises (ohne Umsatzsteueranteil) mit in der Bilanz auszuweisen. Dabei ist jedoch zu beachten, daß das Anlagevermögen über die betriebsgewöhnliche Lebensdauer abzuschreiben ist. Bei der Anschaffung von Hardware geht das Finanzamt in der Regel von einer fünfjährigen Lebensdauer der Geräte aus. In einzelnen Fällen und bei nachweisbaren überdurchschnittlichem Wertverfall kann diese Lebenszeit auch verkürzt werden. Näheres ist hier aus den AfA-Tabellen Ihres Finanzamtes zu entnehmen. Generell läßt sich sagen, daß Hardware, die in den ersten 6 Monaten eines Jahres angeschafft wurde, mit der vollen Jahresrate in Ansatz gebracht werden kann. Hat der Kauf in der zweiten Jahreshälfte stattgefunden, ist der halbe Jahresbetrag anzusetzen. Man hat aber auch die Möglichkeit die Aufteilung nach Monaten vorzunehmen. Dabei ist der volle AfA-Jahresbetrag durch 12 zu teilen und anschließend mit der Zahl der genutzten Monate zu multiplizieren. Eine Besonderheit stellen hierbei jedoch Wirtschaftsgüter von geringem Wert dar. Das Finanzamt sieht solche Anlagegüter, deren Anschaffungspreis unter 800,– DM liegt, als geringwertige Wirtschaftsgüter an. Diese Wirtschaftsgüter können im Anschaffungsjahr in voller Höhe abgesetzt werden.

Zu den Anschaffungskosten zählen grundsätzlich alle Geräte, die die Einsatzfähigkeit des Wirtschaftsgutes ermöglichen. D. h., auch das nachträglich gelieferte oder gekaufte Diskettenlaufwerk, oder der nachträglich angeschaffte Drucker sind in voller Höhe zu aktivieren.

Nun zu den Problemen beim Erwerb der Software und deren Aktivierbarkeit. Der Kauf eines Computers ist immer mit dem Kauf der dazugehörigen Software verbunden. Diese Software ist – wenn der Kauf in DM bewertbar ist – genau wie die Hardware bilanzierungspflichtig und somit in jährlichen Raten abzuschreiben. In der Regel geht man dabei von einer vier- bis siebenjährigen Lebensdauer pro Programm aus. Die reine Softwarebeschaffung unterscheidet sich jedoch von dem Kauf der Hardware mit Software zusammen. Im steuerrechtlichen Sinn wird diese Software als „immaterielles Wirtschaftsgut" angesehen. Damit ist eine steuerliche Abschreibung auf Grund des Bilanzierungsverbotes nicht möglich. Selbsterstellte Programme und Programmanpassungen in gekauften Standardprogrammen, sind nicht aktivierbar. Immaterielle Wirtschaftsgüter sind auch nach dem Berlin-Förderungsgesetz (§ 19 BFG) und auch nach dem Investitionszulage-Gesetz (§ 4 b InvZulG) nicht zu berücksichtigen.

Für den Arbeitnehmer kann man sagen, daß ihn diese Probleme eigentlich nicht berühren. Aber auch er hat die Möglichkeit, den Kauf von Hard- und Software bei der Lohn- und Einkommenssteuererklärung, als Arbeitsmittel unter Werbungskosten, geltend zu machen. Allerdings ist zu beachten, daß die entsprechende Lebensdauer berücksichtigt wird, und das der Erwerb der Hard- und Software zur „Sicherung und Erhaltung" seiner Einnahmen aus nichtselbständiger Arbeit erforderlich ist. Im Klartext heißt das, die Beschaffung dient ausschließlich beruflichen Zwecken.



Archiv-Programme für den Schneider CPC 464

Neu von der DYNAMICS marketing GmbH in Hamburg sind die Archiv-Programme für den Schneider CPC 464; fünf Datenverwaltungs-Programme für einfache unkomplizierte Anwendung: Adressen-Archiv, Bücher-Archiv, Video-Archiv, Schallplatten-Archiv und Artikel-Archiv.

Nachdem die Archiv-Programmserie für den Commodore 64 sich auf dem Markt schnell etabliert hat, entschloß sich die Geschäftsführung, diese praktischen und besonders anwenderfreundlichen Programme auch für den Schneider CPC 464 anzubieten. Konsequent werden alle Funktionen dieser deutschen Software in deutscher Sprache kommentiert. Der Aufbau eines Datensatzes ist fest vorgegeben; ein umständliches Einrichten der Maske der Datei und der Datei-Formate entfällt. Menügesteuerte Editier-, Such- und Druckfunktionen sind selbstverständlich.

Für das „Adressen-Archiv" sind folgende Datenfelder definiert: Name, Vorname, Straße/Nr., PLZ/Ort, Telefon-Nr., Geb.-Datum und „Zusatz".

Das „Bücher-Archiv" zeigt die Datenfelder: Archiv-Nr., Titel, Autor, Art, Inhalt und „Zusatz".

Die Datenfelder für das „Video-Archiv" lauten: Archiv-Nr., Titel, Art, Spielzeit in Min., Zählerstand und „Zusatz".

Für das „Schallplatten-Archiv" sind als Datenfelder definiert: Archiv-Nr., Titel, Interpret, Song, Stil, Länge in Min., Seite der Platte/Liednummer und „Zusatz-Eintragung".

Die Artikel-Nr., der Artikel-Name, Artikel-Ausführung, Bestand, EK-Preis, VK-Preis und die Artikel-Bewertung können beim „Artikel-

(Fortsetzung)

Archiv“ in die vorgesehene Bildschirmmaske eingetragen werden.

Bei der Entwicklung dieser Archiv-Programm-Serie hat man bei DYNAMICS besonderes Gewicht auf eine einfache Bedienung, kombiniert mit einer hohen Leistungsfähigkeit gelegt. Bewußt sollte dem Anwender somit stundenlanges Studium der Bedienungsanleitung und mühseliges Ausprobieren des Programmes erspart werden.
Der Spaß am Computer steht auch beim Adressen-, Bücher-, Video-, Schallplatten- und Artikel-Archiv oben an.

Die Archiv-Programm-Serie für den **Schneider CPC 464** von der DYNAMICS marketing GmbH gibt es ab sofort im Fachhandel zum günstigen Endverbraucherpreis von DM 49,– pro Cassette.

MSX-Arbeitsgemeinschaft gegründet

Ende letzten Jahres gründete eine Vereinigung der Schweizer Vertretungen von Sanyo, Philips, Sony, Yashica, Toshiba, Pioneer, Hitachi und Mitsubishi in Basel die „MSX Arbeitsgemeinschaft Schweiz“. Sinn und Ziel dieser Gemeinschaft, bei der sich erstmals Konkurrenten aus der Unterhaltungselektronik zusammengeschlossen haben, ist es, den 1983 von Microsoft entwickelten MSX Standard für Home- und Personalcomputer in der Schweiz einzuführen. Profitieren soll neben dem Fachhandel, vor allem der private Anwender.

Impressum

CPU
erscheint monatlich im
Tronic-Verlag, 3444 Wehretal 1:

Redaktion:
Axel Credé (verantwortlich)
Rainer Beck, Frank Brall, Siegfried Görk, Hartmut Wendt

Freie Mitarbeiter:
Volker Becker, Holger Grede

Gesamtherstellung:
Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 0 61 21 / 26 60

Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!

Anzeigenleitung:
Annelie Kratzenberg, Heike Lux

Erscheinungsweise:
Erstverkaufstag von
CPU ist Mitte des Monats.

Anzeigenpreise:
Bitte Mediaunterlagen anfordern.

Anzeigenannahmeschluß:
Jeweils 3 Wochen vor Erscheinungstermin.

Urheberrecht:
Alle in CPU veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.
Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.
Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreise:
Einzelheft 5,50 DM
Abonnement: Inland 55,- DM im Jahr (12 Ausgaben)
Ausland: Europa 80,- DM, USA 110,- DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.
Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 100,- DM pro abgedruckter Seite im Heft einverstanden sind.
Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger.
Rücksendung erfolgt nur gegen Erstattung der Unkosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:
Kopierfähige Kassette oder Diskette mit dem Programm (Computer-Bezeichnung), von Drukker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreibmaschinenlistings), evtl. Bildschirmfotos von einem Probelauf und ausführliche Programmbeschreibung (Erklärung der Variablenliste, Beschreibung des Bildschirmaufbaues, Farbe, Grafik usw.). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

IVW geprüft.

Computerriese WANG schloß Vertrag mit der VR China!

Wang, eines der führenden Computerunternehmen für Bürokommunikation, hat drei wichtige Joint-Venture-Vereinbarungen mit der Volksrepublik China unterzeichnet. Die Verträge wurden Ende Dezember 1984 abgeschlossen. Die vorgesehenen Projekte sollen an drei verschiedenen Orten im Norden, im Zentrum sowie im Süden der Volksrepublik China verwirklicht werden.
Das erste mit dem Ministerium für die Elektroindustrie getroffene Abkommen umfaßt ein Joint-Venture zur Produktion von Einstiegssystemen der VS-Computerreihe einschließlich der Entwicklung von Software. Dieser Komplex soll im Laufe der nächsten drei Jahre ein Volumen von rund 50 Millionen US-Dollar erreichen.

Im Rahmen des zweiten Vertrags sollen im Laufe der nächsten fünf Jahre 50 000 Einheiten des Wang-Sekretärinnen-Computers „Büroassistent“ gefertigt werden. Das Abkommen schließt auch die Entwicklung von Software in chinesischer Sprache ein. Nach Ablauf von fünf Jahren wird diese Fertigungsstätte aller Voraussicht nach sämtliche Phasen des Herstellungsprozesses übernehmen.

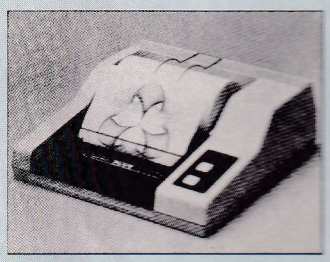
Das dritte Projekt sieht die Montage von 30 000 Einheiten des Wang Professional Computers in dem neuen Industriegebiet von Xiamen vor. Hier werden sehr wahrscheinlich auch weitere Aktivitäten in den übrigen Teilen Chinas ihren Ausgangspunkt haben.

Wang rechnet damit, daß diese drei Projekte aller Voraussicht nach im Laufe der nächsten Jahre ein Umsatzvolumen von mehr als 150 Millionen Dollar erreichen. Am Ende der ersten sechs Monate (31. Dezember 1984) des laufenden Geschäftsjahres FY ’85 wies das Gebiet Asien – Pazifik von Wang das höchste Wachstum im internationalen Bereich überhaupt auf.

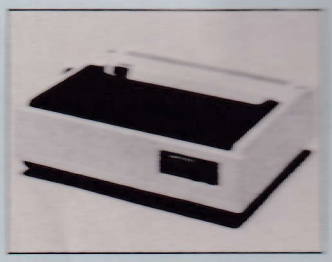
Das Computerunternehmen WANG ist einer der weltweit führenden Hersteller von Bürokommunikationssystemen. Die Produktpalette umfaßt Systeme für die Text-, Daten-, Bild- und Sprachverarbeitung sowie die Telekommunikation und Netzwerktechnik.

Frankreich: MSX User Group gegründet

Gegründet wurde die „Groupe des Utilisateurs MSX“ in erster Linie, um ihren Mitgliedern die Programmbeschaffung für MSX Computer zu erleichtern. Umfassende Informationen und zusätzliche von der Gruppe durchgeführte MSX-Hardwaretests sollen den Anwendern zu Hause den Umgang als auch das Schreiben eigener Programme mit MSX vereinfachen. Wer Interesse an dem Kontakt zu dieser Gruppe hat, kann sich an folgende Anschrift wenden: **Group des Utilisateur MSX, 16 Rue Charpentier 92270 Bois-Colombe.**

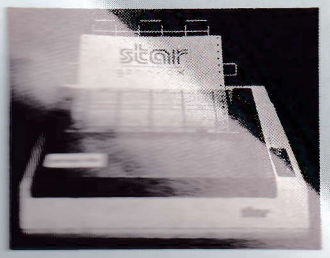


Tandy CGP-115



Silver Reed EXP 400

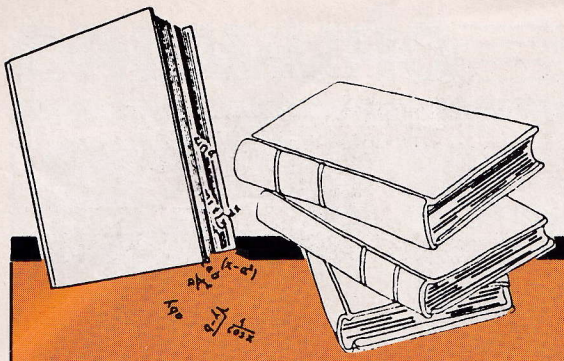
Oki Microline 82 A



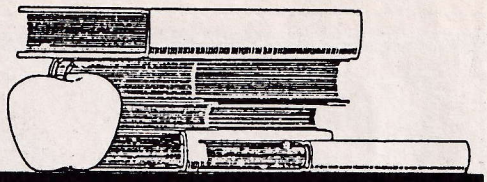
Star Gemini – 10X

Marktübersicht – Drucker unter 1300,– DM

	Modell	Zeichen/Sek.	Zeichen/Zeilen (min./max.)	Typ			Papier			Schnittstellen		Grafik	Pufferspeicher (KB)	Preis DM (inkl. MwSt.) zirka
				Matrix	Typenrad	Tintenstrahl	Endlos	Einzelblatt	Rolle	parallel	seriell			
Juki/Macroton	Speedy 100-80	80	80/142	●			●	●		●		●	2	1000
Juki/Macroton	Juki 2200	10	80/120		●		●	●	●	●	●	●	2	1094
Mannesmann Tally	MT 80	80		●			●	●	●	●		●	2	1300
Mannesmann Tally z. B. Electronic Systems	MT 80	80	40/132	●			●	●	●	●		●	1 Zeile	1300
Oki	Microline 82 A	120	132	●			●	●	●	●	●	●		1254
Robotron/Unitronic	Europrint K 6316	100	80	●			●	●	●	●	●	●	2 Zeilen	890
Robotron/Unitronic	K 6311	100		●			●	●	●	●	●	●		890
Shinwa/ C. Melchers	CPA-80			●			●	●	●	●		●	1 Zeile	1000
Shinwa/ C. Melchers	CP-80	80		●			●	●	●	●		●	1 Zeile	900
Shinwa/ C. Melchers	CP-80X	80		●			●	●	●	●	●	●	1 Zeile	1000
Star	Gemini-10X	120	80/136	●			●	●	●	●		●	1 Zeile	1195
Star	STX-80	60	40/80	Thermo					●	●		●	1 Zeile	595
Silver Seiko/ Silver Reed	EXP 400	12	82/98		●		●	●		●		●		1140
Smith Corona/SCM	D-100	120	80	●			●	●	●	●		●	1 Zeile	1140
Smith Corona/SCM	D-80	80	80	●			●		●	●		●	1 Zeile	800
Tandy	TP-10	30	32	Thermo			●	●	●		●	●		300
Tandy	CGP-115	12	40/80			●			●	●	●	●		645
Tandy	DMP-110	50	136	●			●	●		●	●	●	80 Zeichen	1195
Thomson Electronic	PR90–040		40/320	Thermo			●		●	●		●	40 Zeichen	750
Thomson Electronic	PR90–080	50	80	●			●		●	●	●		90 bit	750
Uchida/Synelec	DWX-305	20			●		●	●		●	●	●		1300



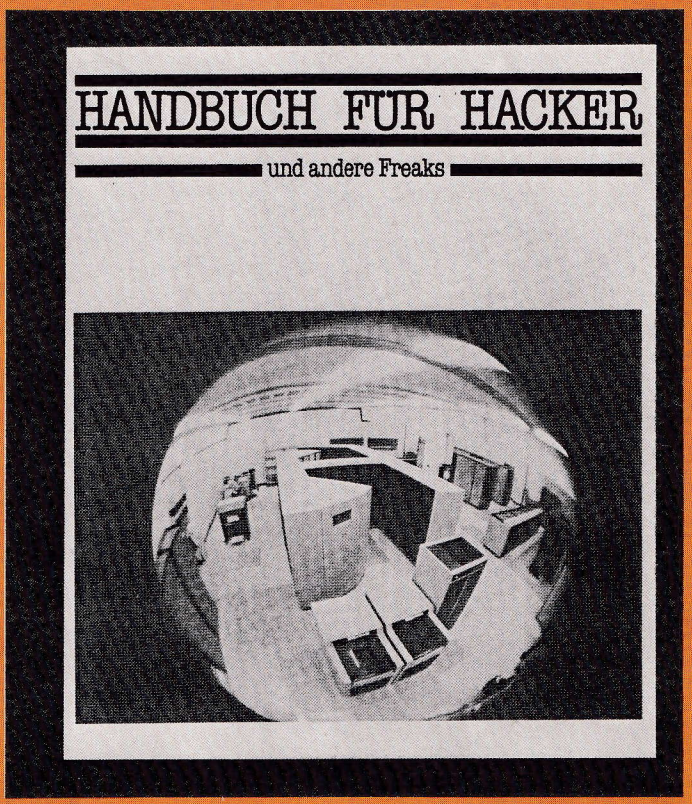
CPU Bibliothek



Mikro-Computerbenutzung in der betrieblichen Praxis

Ohne viel Aufwand wird Statistik handlich. Der Anwender kann im Wirtschafts-Alltag Ergebnisse erhalten, die wesentlich zur unternehmerischen Entscheidungsfindung beitragen. Das im GABLER Verlag, Wiesbaden, erschienene Buch der Professoren Kurt Scharnbacher und Gustav Kastner, **Statistik im Betrieb mit Basic auf Commodore,** DM 39,50, bietet 45 an der Praxis orientierte Basic-Programme für Mikro-Computer an. Die Praxisbezogenheit des Buches zeigt sich daran, daß die Programme auf den meisten Mikro-Computern mit Basic-Interpreter ablaufen können. Durch minimale Modifikationen lassen sich die Programme problemlos an andere Betriebssysteme anpassen. Auf Wunsch sind die Programme auch als Diskette erhältlich.

Gabler Verlag, Wiesbaden



Ein Handbuch fürs Hacken

Geht das überhaupt? Ein Buch, das zur (unbefugten) Kommunikation mit fremden Rechnern anleitet und auf dunklen Kanälen vertrieben wird? Die Post hat jedenfalls gleich eins der ersten Exemplare des jetzt in Hannover im Selbstverlag erschienenen „Handbuch für Hacker und andere Freaks" erstanden – wurde aber nicht fündig.

Denn die hiesigen Elektronik-Fans, die seit eineinhalb Jahren in Heim-Computer-Arbeit die vertraulichen Tips der Hacker sammeln, haben sich für ihr Handbuch etwas einfallen lassen: Es wird als Loseblattsammlung in mehreren Teilen geliefert. In Buchhandlungen kaufen und unter der Bezugsadresse bestellen kann man bis jetzt nur das erste Drittel, das auf 90 Blättern das juristisch völlig unbedenkliche Grundwissen enthält, das der Computer-Fan braucht, um mit der Hackerei erst einmal beginnen zu können. Wenn der Käufer später seine Adresse mitgeteilt hat, werden dann die beiden anderen Lieferungen irgendwann bei ihm im Briefkasten liegen.

Damit kauft man zwar quasi die Katze im Sack, doch der erste Teil enthält immerhin genau das, was sich der Jung-Hacker ansonsten nur mit Mühe aus Zeitschriften und Infos zusammensuchen könnte, und vieles mehr.

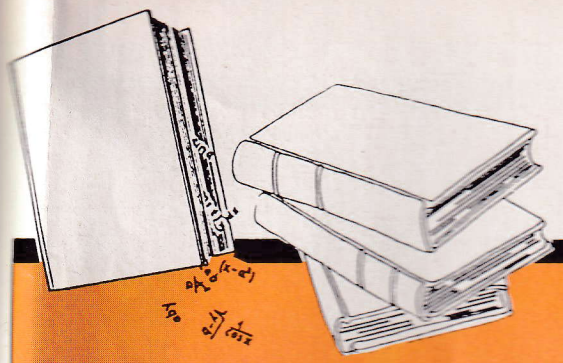
Beschrieben werden die Datennetze, was der Hacker an Ausstattung braucht und wie man erfolgreich mit anderen Computern in Verbindung tritt.

Die nötige Ausstattung besteht aus einem billigen Heimcomputer, einem Selbsbau-Akkustikkoppler, einem Telefonanschluß und dem Programm. Es sollte eigentlich für den normalen Elektronikfreak ohne weiteres möglich sein, diese wenigen Gerätschaften fachgerecht zusammenzustecken. An dieser Klippe scheitern bereits sehr viele potentielle Hacker in der Praxis, denn trotz der Normierung paßt kein Gerät zum anderen, obwohl die Werbung und das Aussehen der Stecker es versprechen. Hier hilft das Handbuch mit dem Kapitel „RS-232-C Schnittstelle".

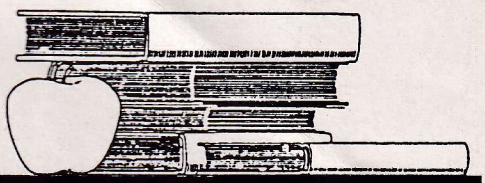
Es geht um die praktischen Tips: Wo bekomme ich die geeigneten Programme her (gegen Leerdiskette und Rückporto kann man sie auch bei den Verfassern beziehen), wie komme ich in die Datennetze (bekannte und unbekannte wie z. B. 0131-165) und kann dabei noch Gebühren sparen!

Selbstverständlich fehlen auch die Nummern der Hakker-Mailboxen (Mb) nicht, über die vertrauliche Informationen anonym ausgetauscht werden. Für den Laien: Eine Mailbox ist ein über das öffentliche Telefonnetz anwählbarer privater Kleincomputer, der seinen Speicher als „Schwarzes Brett" anbietet.

Wer sich über das Datex-P-Netz der Post beispielsweise in den US-Air-Force-Computer einwählen will, muß folgendes beachten: seinen Koppler auf 300 Band und Vollduplex einstellen. Das Terminalprogramm sollte auf 8 Datenbits, 2 Stopbits, keine Parität und Vollduplex stehen – man sieht, das Buch ist nicht für den völligen Anfänger geschrieben –, doch wählen kann die Nummer jeder. 32.66.51 (beispielsweise aus Hannover), um in das



CPU Bibliothek



Datex-P-Netz zu kommen, dann nach der Netzmeldung 456.1104.0250. Wenn man „USAF" und „AIRFORCE" eingetippt hat, sind die Barrieren „Identifizierung" und „Password" schon überwunden. Das Handbuch liefert nicht nur seitenweise die zwar nicht geheimen, aber auch nicht öffentlich erhältlichen Nummern von Computern, sondern auch das Schema, mit dem man aus dem Standort eines bundesdeutschen Computers dessen zwölfstelligen Anschluß bis auf drei Ziffern rekonstruieren kann. Für die letzten drei Ziffern gilt dann die Methode Versuch und Irrtum.

Dieser erste bisher erschienene Teil des Handbuchs hilft auch noch mit vielen weiteren Tips über die Hürde hinweg, die vor den Informationen in anderen Systemen stehen. In den Nachlieferungen kommen die Nummern aller bundesdeutschen Computer und unter anderem Hilfreiches für Datenreisende im BTX-System.

Ein Haken bleibt allerdings bei diesem Handuch: Durch die gezwungenermaßen umständliche Aufmachung und Vertriebsform kostet es **38 DM (incl. Versand, Nachlieferungen und Hackermailbox-Zugriff).**

Zu beziehen bei Regine Rathmann/Jürgen Schalla, Friesenstr. 24, 3000 Hannover 1.

Mikro-Computer in der Betriebspraxis

Dem Anwender in Klein- und Mittelbetrieben dient das soeben im Gabler Verlag, Wiesbaden erschienene Buch von Gustav Kastner **Operations Research mit BASIC auf Commodore.** DM 47,–, das eine themenbezogene Programmsammlung zum Bereich des Operations Research enthält. Neben der Problemstellung wird schrittweise die Entwicklung der Programmlogik und die daraus resultierenden Programm-Listings mit konkreten Beispielen angeführt. Durch minimale Modifikationen lassen sich die Programme problemlos an andere Systeme anpassen. Bei den meisten Programmen ist eine Datenausgabe über Drucker vorgesehen, ohne daß diese obligatorisch ist.

Gabler Verlag, Wiesbaden

Wenn Sie in der Lage sind, auch andere an Ihren Phantasien teilhaben zu lassen, versuchen Sie es doch einfach mal mit selbsterstellten Programmen!

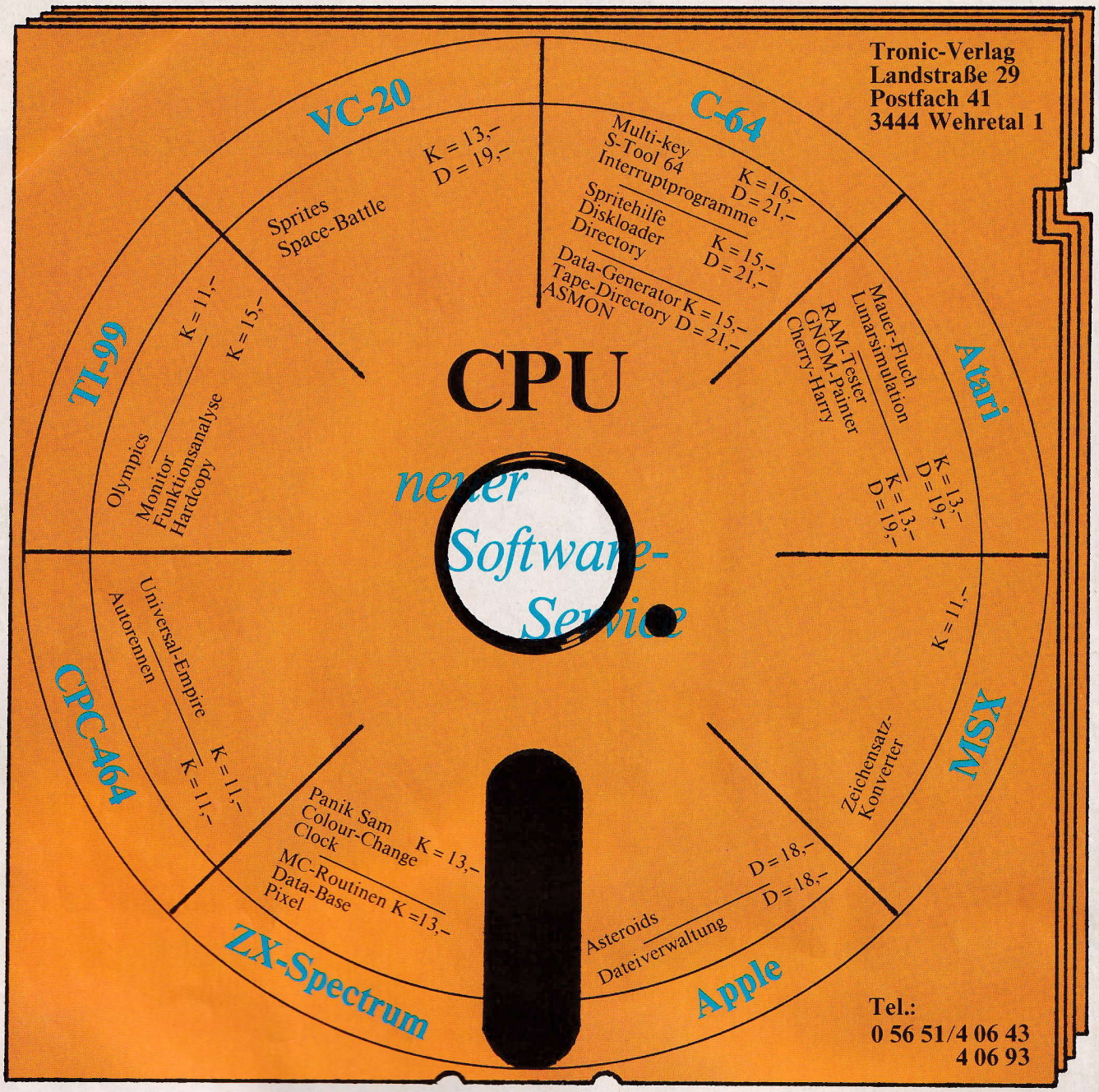
Wir suchen ständig neue, phantastische Abenteuer, spannende und lustige Programme und natürlich auch Anwendersoftware.

Wir brauchen von Ihnen:

1) Ausführliche Spielbeschreibung
2) Datenträger (Kassette oder Diskette)
3) Komplettes Listing (nicht unbedingt erforderlich)

All dies schicken Sie bitte an den Tronic-Verlag, Postfach, 3444 Wehretal 1.

Wir zahlen: 120,– DM pro abgedruckte Seite



CPU Bestellkarte-Softwareservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service über den Verlag bezogen werden.
(Ausland nur gegen Vorkasse)

Die Zustellung erfolgt: gegen **Vorkasse** ☐ oder Inland per **Nachnahme** ☐
+ Versandkosten

innerhalb von 1 Woche

Entnehmen Sie bitte aus unseren Preislisten die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir:

☐ Cassette für _____ (System) ☐ Anzahl

☐ Diskette für _____ (System) ☐ Anzahl

zum Preis von gesamt _____ DM

Name/Vorname: Straße, Nr.:

PLZ/Ort: Datum, Unterschrift

BILLIGER

wird auch Ihre Homecomputerzeitschrift

wenn Sie sich für ein Jahresabonnement
von
HOMECOMPUTER, COMPUTRONIC
und CPU als Kombination
entscheiden.

Für
6 Hefte von HOMECOMPUTER,
6 Hefte von COMPUTRONIC und
12 Hefte von CPU
zahlen Sie im Jahresabonnement
statt 141,– DM nur 100,– DM.
Sie sparen also sage und schreibe: 41,– DM.

Bei Lieferung in das europäische Ausland beträgt
das Jahresabonnement HOMECOMPUTER, COMPUTRONIC
und CPU 150,– DM, nach Übersee 200,– DM

Wir garantieren:

★ Sie erhalten HOMECOMPUTER, COMPUTRONIC und CPU ab der nächsterreichbaren Ausgabe.
★ Lieferung erfolgt frei Haus inkl. Mehrwertsteuer.
★ Die Zustellgebühren sind im günstigen Abonnementpreis enthalten
★ Es entstehen Ihnen keine weiteren Kosten.

Ausschneiden und einsenden an:

Verlagsunion
Friedrich-Bergius-Straße 20
Postfach 57 07
6200 Wiesbaden

Bestellgutschein

Ja, ich möchte ins Computerleben einsteigen und bestelle deshalb:

☐ Jahresabonnement **Kombi: HC, Computronic und CPU** (24 Hefte) 100,– DM
☐ Jahresabonnement **HC und Computronic** (12 Hefte) 55,– DM
☐ Jahresabonnement **CPU** (12 Hefte) 55,– DM

Name/Vorname

Straße, Nr. PLZ, Ort

Ich wünsche folgende Zahlungsweise:

☐ Bargeldlos durch Bankeinzug
Bankleitzahl

Geldinstitut Konto-Nr.

☐ gegen Rechnung

Datum, Unterschrift

Abonnements-Kündigungen:
6 Wochen vor Ablauf des Jahresabonnements.
Außerdem kann ich diese Bestellung innerhalb von 10 Tagen beim Verlag widerrufen!

Datum, Unterschrift

magna

HOME-COMPUTER CASSETTEN + DISKETTEN

CASSETTEN

Präzise Cassetten-Mechanik
Hohe Speicherdichte
Für alle Data-Recorder

DATA-DISK DISKETTEN

Extreme Lebensdauer durch zusätzliche Oxygenbeschichtung

Zuverlässige Datensicherheit durch mehr als 70 chemische, magnetische und elektrische Qualitäts-Tests

magna tonträger vertriebs gmbh
Bunzlauer Straße 3 · Postfach 400340 · 5000 Köln 40
Telefon (02234) 74054 · Telex 889975

TONTRÄGER